**Web Appendix for**

# Investigating the Impact of Airbnb's Smart Pricing Algorithm on Racial Disparities across Airbnb Hosts

## Table of Contents

## Section 1. Technical Notes on Facial Analyses

The key individual demographic variable—namely ethnicity and age—are not provided/available on Airbnb. Hence the only option was to obtain such information from the profile photos of the hosts. Given the large number of hosts in our sample, we leverage the advances in Convolutional Neural Networks (CNNs, an emerging deep learning framework, see Krizhevsky et al. 2012 and Simonyan and Zisserman 2015) to predict the ethnicity and age in a scalable and automatic way. Specifically, we employ the framework of ResNet-50, a CNNs 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, given any Airbnb hosts' photo, his/her ethnicity and age. To do so, we first construct a large data sets that consist of human face photos, each labeled with the ethnicity and age. Next, we train the deep learning model (i.e., ResNet-50 in this paper) on the training set. The deep learning model is optimized to extract facial features from face photos and learn the relationship between the facial features and the corresponding labels. Finally, the trained classifier is applied on our Airbnb face photos to predict the labels for each host in our sample. Below we describe each step.

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

For ethnicity classification, we combine multiple public face databases, including 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 age prediction, we use IMDB-WIKI image database, which is constructed of 0.5 million images of celebrities crawled from IMDB and Wikipedia webpages. The date of birth of each celebrity combined with the date that each photo was taken output the age labels (Rothe et al. 2015).

### C. Preprocessing: detecting and extracting faces

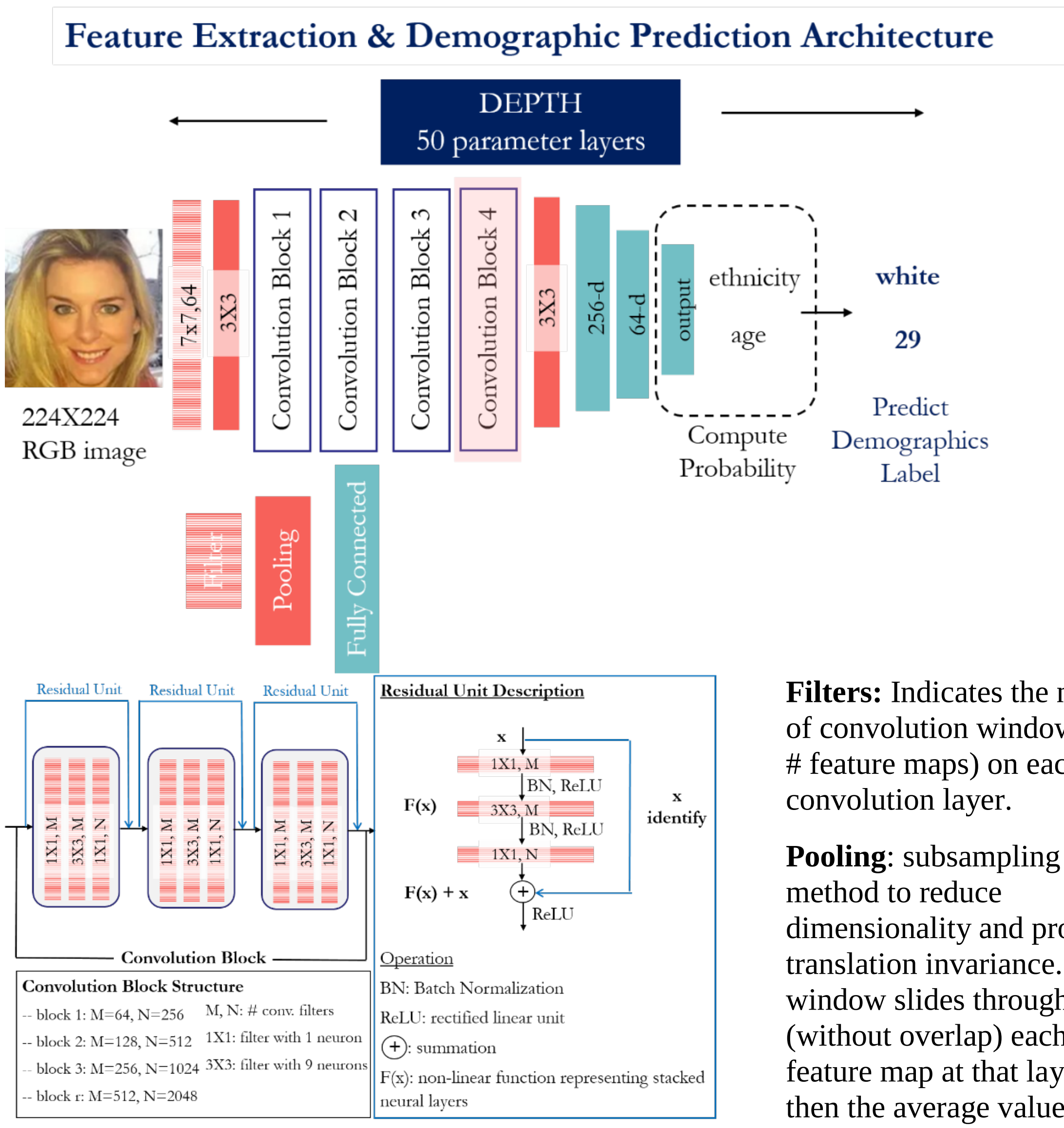
Our Deep learning model analyzes the face photos. However, some images contain content that does not belong to a person's face, e.g., body, shoulder, background. Hence, for each image, we first detect the existence of a face, then extract the face. For images in the training set (i.e., the public face data), this is straightforward since each image contains one face. For images in the Airbnb sample, very few images do not photograph any person's face, hence were discarded. For images that contained multiple faces, we extracted and stored all faces for analysis.

A CNNs 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. Each layer is basically a multidimensional matrix, with each neuro 'carrying' a weight that represents the

[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/.

numeric value of each element. The number of layers that carry weight define the ‘depth’ of a deep learning model.

**Figure S1 The Architecture and Layer Description of ResNet-50. Shows the training framework, architecture of the deep learning model, and the detailed description of layer operations.**




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

**Pooling**: subsampling method to reduce dimensionality and provide translation invariance. A 3x3 window slides through (without overlap) each feature map at that layer, and then the average value in the window is picked as representation of the window.

In a deep learning framework, high dimensional data such as images and texts are expressed as multidimensional matrices/arrays. Then the model processes the data through the neuron layers implementing matrix multiplication on the data. What defines a CNNs is a special layer—convolution layer, which operates dot productions on the input data (below we will describe operation of convolution layers)

Multiple architectures of CNNs have been proposed, including AlexNet by Krizhevsky et al. (2012), VGG by Simonyan and Zisserman (2015), and Inception/GoogleNet by Szegedy et al. (2015), and ResNet (He et al. 2016). These variants follow classic CNNs framework—consisting sequence of convolution layers) but differ from each other on number/order/size of the layers in the sequence. Some advanced model such as Inception/GoogleNet and ResNet implemented special computation units in the model to tackle down particular challenges in the training of CNNs.

In this study, we used ResNet-50—a ResNet model containing 50 parameter layers—since it has been shown to effectively solve the challenge that the gradient vanishing problem makes a very deep CNNs difficult to train (Glorot and Bengio 2010). This intriguing framework propose by He et al. (2016) has quickly gathered attention because the introduced residual learning functions make it easier to optimize a very deep neural network. ResNet provides 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 on the architecture of ResNet-50[5].

**D. Architectures of ResNet-50**

Figure S1presents the architecture of ResNet-50. As can been seen, the model is constructed of 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 images (i.e., the face photo of a person). An image is simply a 3-d matrix, with its weights equal to the pixel intensity in the 3 channels (RGB). All images (both in training and in prediction tasks) are resized to 224X224 (in pixel) be consistent with 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. This operation generates an intermediate output (also represented by a multi-dimensional matrix), which can be viewed as 'useful information' extracted from the image and serve as the input for the next layer. Such implementations continue till the last layer of the model, i.e., the output layer that computes the probability distribution over the multiple labels. The probability distribution is then converted to labels.

Take ethnicity prediction as an example. For each possible label $l$ ($l$= {*white*, *black*, *others*}), the output layer, given the input it receives $X$ and the its neuron weights $W_1$ and $W_0$, computes:

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

where $L$ 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., (he 3-d fully connected layer). $X^T$represents the output from the layer preceding the output layer. Then, for any image $IMG^k$ in the training set, the model outputs the label with the highest probability:

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

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

Note that $X$ is the output extracted via the implementations on all the preceding layers on the input image, hence it is a function of the model weights and the input image, i.e. $X = \Phi(IMG^k, W)$. For each training image $IMG^k$, we know its true label, hence we can optimize the model by adjusting the weights such that it predicts labels as accurate as possible.

Throughout the CNN model, there are a sequence of such weights on each layer, and the weights define the intermediate extracted vectors from each layer, including $X^T$as described above. Theses weights are adjusted during the training process, so as to optimize the model's performance on predicting the images in the training set.

## E. Operations of key layers

We describe convolution layer and pooling layer, which are the key layers in a CNNs.

### Convolution layer

The convolution layer is the most important and unique layer in the CNN. A convolution layer consists of a stack of so-called convolution filter or convolution kernel. A convolution filter is simply a matrix with each element representing a numeric value. For example, in a convolution block, a convolution layer with a size of 3X3 and hence consists of 9 such numeric values[6]. Such a matrix, treating an image or an intermediate input as a matrix, operates a dot production by 'sliding' through the input. Therefore, for an input with relatively large size (e.g., 224X224), a 3X3convolution filter operates dot production for every 3X3 patch on that input matrix. The nice features of convolution operation are that: 1) it reduces the dimensionality of parameters, and 2) it well explores and reserves the (local) spatial relationships of the input. Particularly, an intuitive example of the second feature is that: if a convolution kernel extracts a particular oriented edge of an object, then operating this kernel on every small square (e.g., 3X3 and 1X1) on an image would extract all edges in that direct from the image. Many of such kernels that extract edges would extract edges in all directions—potentially constructing the contour of an object. As can been seen in Figure S1, each of the blocks consist of varying numbers of convolutional filters (e.g., 64, 128, 256, 512, 1024, and 2048 filters). Hence, these kernels extract features from an input data, which represents the extracted features from the preceding layers. Towards the output layer in the CNN, the filters combined extract higher- and higher- level features. That is, the CNN is able to extract a hierarchical structure of features that are related to predict the output labels.

### Pooling layer

It's a common practice in CNN to insert a pooling layer in-between the successive convolution layer. A pooling layer is a small square filter. In our model, the pooling filter is a 3X3 matrix. Similar to the operation of convolution filter, an average-pooling layer applies to every 3X3 square patch on an input data. The function of a pooling layer is to pick and using the average value in that 3X3 square. Adding pooling layers can reduce the spatial size of the intermediate features and

[6] The size of a convolution layer is a choice of the model architecture. 3X3 is a widely-used choice. Another common choice is 5X5.

the dimension of the trained parameters in the model. Particularly, it helps to efficiently prevent the problem of over-fitting.

**F. Training technical notes**

The three prediction tasks are implemented independently. For each task, we first randomly split the corresponding dataset, with 80% of the samples form the training set and the remaining 20% form the (hold-out) test set. To effectively learn facial features that have predicative power on ethnicity and age, we leverage transfer learning and build our model on top of an existing deep learning model that was well-trained for a related task. Specifically, we adopted the model of Cao et al. (2018), which trained a ResNet-50 on a large-scale face dataset, VGGFace2 that contains 3.3 million images for over 9,000 subjects[7]. That is, our demographic classification model was built on a classic ResNet-50, with slight architecture modification. Specifically, the output layer in the original RexNet-50 was removed as it was specific to the original task (object classification). We then add three fully connected layers on top of that (dimension of 256, 64, 3 respectively), where the last layer is output layer.

To improve the training process, we initialize the model weights with the pre-trained weights of the original ResNet-50 and then fine-tune the parameters. For images, the extracted information is generic, to some extent, across various tasks (e.g., early layers in CNNs serve as edge and contour detectors). Hence, we were able to optimize our model starting from a point where it was already close to 'optimum'. The pre-trained RestNet-50 model was optimized for facial recognition task. Hence, we efficiently improved the learning process of our model, with the initialized able to extract facial features, from the images, that are relevant to the identity of a subject. The added three layers, without pre-trained weights available, were initialized with LeCun's uniform scaled initiation method (LeCun et al. 1998).

To improve the generalization power of the trained model, we employed a real-time data augmentation method, by randomly flipping, rescaling, and rotating the training samples during the training process (Krizhevsky et al. 2012). Specifically, we implement a real-time (i.e., during training) image transformation over each image in the training sample, by randomly 1) flipping input image horizontally, 2) rescaling input image within a scale of 1.2, 3) rotating the image within $20°$. This method introduces random variation in the training sample, increasing the training set size and reducing the overfitting.

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

The ethnicity classifier achieved an average accuracy of 92.5% (for classifying a person into three categories—white, black, and others). The age classifier achieved a MAE (mean absolute error) of 4.750 (for predicting a person's age between 1 and 100).

Once the RestNet-50 model was optimized to learn the relationship between the facial features and the image label. We then perform ethnicity-age predictions for the Airbnb hosts in our sample. Specifically, for each type of predictions, the trained ResNet-50 model extracted the facial features, from each face image, that are effective to predict the label. The model then assigns a label to the person, based on the extracted facial features and the learned feature-label relationship.

---

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

## Section 2. Data Sources, Descriptive Statistics and Anecdotal Evidence from Host Forums

### 2.1 Data Sources

We briefly discuss the main data sources that we have used in our paper and the variables that we obtained from each data source.

(i) InsideAirbnb.com: This is a publicly available website from which we got the list of all Airbnb listings in each of the seven cities (Austin, Boston, Los Angeles, New York, San Diego, San Francisco and Seattle) and the unique listing IDs associated with each of the listings. As we will discuss in section 2.2, we used this list to randomly sample the hosts that we study in our analysis.

(ii) AirDNA: We used AirDNA data to obtain information on average prices per month, occupancy status per month, and other characteristics mentioned on webpages of each property. This data spans from July 2015 to Aug 2017, and it covers all Airbnb listings in the seven cities (Austin, Boston, Los Angeles, New York, San Diego, San Francisco and Seattle).

(iii) Scraping: We scraped the websites of all properties in our random sample to get daily information on whether or not the host used smart pricing algorithm, and also to get the profile photos of the hosts from which we inferred their ethnicity and age.

(iv) Finally we used Zillow Research for inferring the values of the listed properties and median earnings and the education level in the neighborhood from American Community Survey.

### 2.2 Sample Construction

We first obtained the data from AirDNA, which spans from July 2015 to Aug 2017, and it covers all Airbnb listings in the seven cities (Austin, Boston, Los Angeles, New York, San Diego, San Francisco and Seattle). This data has information on average monthly prices, monthly occupancy status, and all other characteristics mentioned on webpages of the properties.

Our next task was to randomly sample a set of properties from the entire set of Airbnb properties. To do so, we obtained list of all Airbnb listings in each of the seven cities from a publicly available website, viz., InsideAirbnb.com. There were a total of 66,424 properties in this list as of October 2015, and each listing was associated with a unique ID. We randomly selected 13,200 properties (roughly 20% of the listings IDs) from the overall list. We used Python's library, '*random.sample,'* to shuffle and randomly sample the 13,200 properties (roughly 20% of the listings IDs) from the overall list. The library *random.sample*(*population, k*) takes two inputs: *population* is the list or sequence from which the a random sample is chosen, and *k* is the length of returned sample list, that is, the number of random elements to choose from *population*. This random sampling was done without replacement. Calling *random.sample*(*population, k*) returns a new list containing *k* elements, which are randomly sampled from *population*. In our implementation of the library *random.sample*(*population, k*), we set *population* = the list of 66,424 property IDs and set *k* = 13,200. The implementation thus returned a list of *k* = 13,200 randomly selected property IDs.

Next, we discuss how we used the random sampled IDs to get to our final estimation sample. For each randomly selected property, we scraped its website to get its property calendar at the end of each month starting from Nov 2015. Out of the total of 13,200 properties, 12,587 properties returned a valid calendar page. The remaining listings returned an invalid page or an error. Thus we did not consider these for our analysis. 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 algorithm.

For each of these 12,587 properties, we collected hosts' profile photos by scraping the posted host photos on the webpages. Out of these, 10,924 properties had a host photo from which a human face was detected. As we discussed earlier, we used the hosts' photos to infer their ethnicity and age. We did not consider the remaining properties for further analysis because they had photographs of either non-human objects (pets, background etc.) or the photographed face was too unclear and small to make a judgment on the person's ethnicity or age. We then matched the listing of these 10,924 properties to the properties in the AirDNA data. Out of the 10,924 properties, we were able to match 10,903 properties with the properties in the AirDNA data.

Our final step was to deal with the 'stale vacancies' issue. A stale vacancy issue refers to a scenario where a property is listed but the host neglected to update the listing status. As a result, a property may appear to be available but would never be reserved because hosts did not respond to any booking request. Fradkin et al. (2017) found that about 15% of the time that guest requests were rejected due to this issue. To address this issue, following Zalmanson et al. (2018) we removed properties that did not have any booking one year prior to the natural experiment and throughout the whole observation window. This resulted in 9,396 properties, which represented our final sample.

We next describe the variables and the summary statistics in the final sample.

## 2.3 Data description

### *Full statistics of variables*

In Table S1, we report the statistics of the variables in our sample, grouped by adoption decision.. The data represents the sample after excluding properties to address the stale vacancy issue and before we applied IPTW strategy (inverse probability of treatment weighting). There are 9,396 properties, out of which 2,118 properties are adopters.

**Table S1 Sample statistics: grouped by adoption. Shows the full statistics, presented by adoption group.**

| | (1) Adopters | | (2) Non-adopters | | (3) All Properties | |
|---|---|---|---|---|---|---|
| **VARIABLES** | **Mean** | **Std. Dev.** | **Mean** | **Std. Dev.** | **Mean** | **Std. Dev.** |
| # Properties | 2118 | | 7278 | | 9396 | |
| **Airbnb Host Demographics (Measured from Face Data)** | | | | | | |
| White (Ethnicity) | 0.83 | 0.37 | 0.75 | 0.43 | 0.77 | 0.42 |
| Black (Ethnicity) | 0.07 | 0.25 | 0.12 | 0.32 | 0.11 | 0.31 |
| Others (Ethnicity) | 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 (%) | 51.36 | 19.28 | 54.42 | 19.06 | 53.68 | 19.15 |
| Median Home Earning (*1000 USD*) | 48.88 | 20.52 | 52.11 | 22.04 | 51.33 | 21.73 |
| **Airbnb Property Amenities Information (all variables are binary features)** | | | | | | |
| 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 |

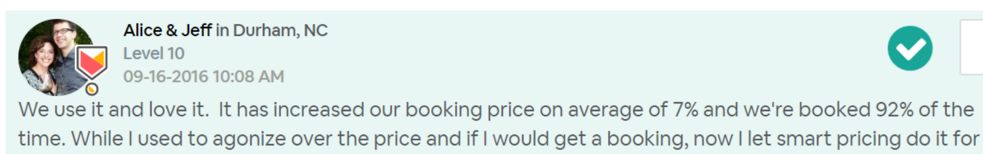
| | | | | | | |
|---|---|---|---|---|---|---|
| Laptop-friendly | 0.52 | 0.50 | 0.30 | 0.46 | 0.35 | 0.48 |
| Fireplace | 0.13 | 0.34 | 0.13 | 0.34 | 0.13 | 0.34 |
| Elevator | 0.16 | 0.37 | 0.21 | 0.41 | 0.20 | 0.40 |
| Gym | 0.06 | 0.23 | 0.08 | 0.27 | 0.07 | 0.26 |
| Family-friendly | 0.15 | 0.36 | 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 |

### 2.4 Anecdotal evidence on reasons to adopt/not adopt smart pricing algorithm

Airbnb hosts self-select into adopting the pricing algorithm. To better understand their selection process, we identified the primary reasons for adoption/non-adoption of the algorithm from the Airbnb host forums. In Table S2 we present a piece of anecdotal evidence that we collected for the three main reasons for adoption/non-adoption decisions—1) the host may want to adopt the algorithm if his/her cost of frequently changing prices is high, and 2) a host may not want to adopt algorithm if the algorithm-recommended prices are very low 3) a host may not adopt because s/he does not trust the algorithm or does not trust Airbnb's motives.

**Table S2 Evidence from host forums – factors that may affect a host's adoption of smart pricing. Shows examples of discussions from Airbnb hosts regarding the reasons they adopted/not adopted Airbnb's smart pricing algorithm.**

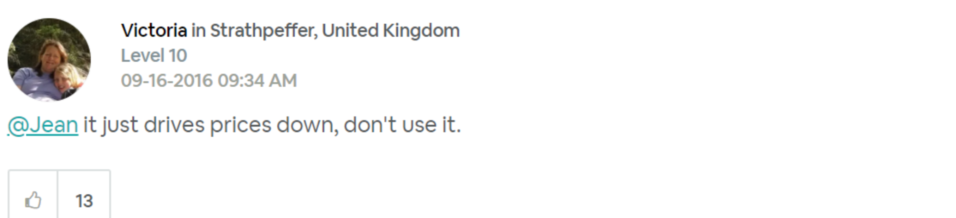
*Category 1*


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 for

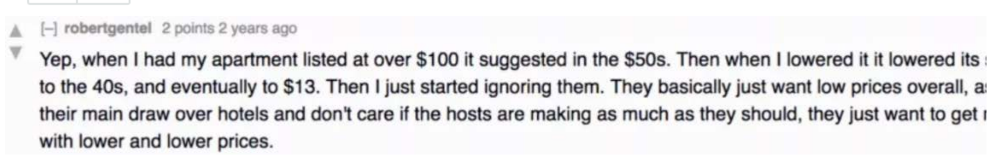
*Category 2*


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, a
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.

Building on our research on the Airbnb host discussions on their reasons for adopting/not adopting the algorithm (Table S2), we summarize these reasons into two categories. Below we discuss the measures that we used to capture the two categories of reasons.

***Category 1***: High opportunity cost of time: hosts with a high opportunity cost of time have a greater incentive to adopt since they then do not have to expend effort to decide and update prices every day. We include the following variables to capture this cost: (i) $MedianEarning_{in}$ and $HomeValue_i$. $MedianEarning_{in}$ is defined as the median earnings in a neighborhood n conditioned on host i's ethnicity and age. We collected this information from the American Community Survey[8]. $HomeValue_{in}$ is defined as the average home value across homes in neighborhood n with the same size as that of host *i*. We collected these data from Zillow Research[9]. (ii) How long has the property been listed 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 are more experienced, own multiple listings, and manage entire places are the ones who as such dynamically update their prices over time, and thus have lower costs of updating prices.

***Category 2*:** This relates to the reasons for not adopting the algorithm, which are: *a*) lack of trust in the technology and in Airbnb (hosts believe that Airbnb does not have their best interests in mind), and *b*) prices recommended by the algorithm are too low. We include the following variables to capture these reasons: (i) age and education level of hosts, which capture their receptiveness towards the new technology. The education level is captured by $Bachelor_{in}$ and $Graduate_{in}$ which are the ratios of people in the same neighborhood *n* who have a bachelor's and master's degrees respectively and share the same demographics as host *i*. (ii) Whether the property has unique features. This captures reason (b) - while the algorithm takes standard features into account when pricing a property, it may not incorporate the more unique features (Ye et al. 2018), and hence may set a lower price for such properties. To capture this effect, we include the presence of all features and amenities listed on the property webpages.

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

We used AirDNA data to obtain information on average monthly prices, monthly occupancy status, and other characteristics mentioned on webpages of the properties. This data spans from July 2015 to Aug 2017, and it covers all Airbnb listings in the seven cities. 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 a Month = Average Nightly Price × Occupancy*

*Occupancy= # booked days in a month/(Total # of days in a month - # blocked days in a month).*

In the formula above, the number of blocked days in a month are the number of days in which the property not available to any guest because it was blocked by the host (for say personal use), and the number of booked days in a month are the number of days in which the property was rented out/reserved by guests. The above formula shows that in order to compute occupancy and revenues, we need to know the prices, number of booked days in a month and the number of blocked days in a month. 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 Dec 2015, AirDNA was able to use Airbnb's API to get accurate information

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

[9] https://www.zillow.com/research/data/.

on the number of booked as well as blocked days in a month for each property. However from Dec 2015 onwards, Airbnb disabled its API. As a result, AirDNA could only observe the sum of the number of blocked and booked days in a month, but not their individual values. To deal with this, AirDNA built a proprietary ML algorithm that helps determine the number of blocked vs. booked days for each property in each month.

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 booked or blocked. AirDNA provides two reasons why its algorithm is fairly accurate in its predictions reasons. First, the data that AirDNA uses for training and prediction testing of its ML algorithm is of high quality obtained from two sources[10].

(i) Historical data that AirDNA had 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) The ongoing stream of data that AirDNA has been collecting since Dec 2015 through its collaborations with individual hosts and professional managers who manage large number of listings on behalf of Airbnb hosts. Based on our conversations with AirDNA, there are more than 650,000 such properties under collaborative arrangement across the U.S. and Europe. From these individual hosts and professional managers, AirDNA gets daily information on each property's true reservation status (i.e., whether it was unreserved, booked or blocked), daily prices and other property characteristics. The hosts and managers also provide useful insights to AirDNA on what other behavioral variables they should feed into the ML algorithm to improve its predictability.

Both these sources of data contain rich set of explanatory variables (such as location, time, property characteristics, length of booking, booking lead time, historical performance of the property, and other behavioral variables learnt from hosts and professional managers etc.) which AirDNA feeds into their algorithm. Moreover, since the second source of data mentioned above is an ongoing stream of data, AirDNA's ML algorithm continues to learn and improve as time goes on.

*Second*, 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 predicting accuracy on their test set across time, which includes periods before and after Dec 2015.

[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

Inverse Probability Treatment Weighting (IPTW) is a widely-applied weighting method to construct a balanced sample of treated and untreated units (Rosenbaum 1987, Austin and Stuart 2015). This method first computes the weight for each given unit as the inverse of its treatment probability, and then weighs all units to eliminate any existing systematic differences between treated and untreated units in terms of the observed covariates that explain the treatment probability. Unlike matching methods in which units in the two groups that are not very similar to each other are discarded, the IPTW method does not have to discard any unit when creating a balanced sample (Guo and Fraser 2015). Prior Monte Carlo studies have shown that IPTW method leads to lower mean squared error in the estimates of treatment effects as compared to those obtained from matching methods such as Propensity Score Matching (Austin 2013, Austin and Stuart 2015).

In what follows, we first provide an overview of the IPTW method and explain how it works. Following that we will explain each of the steps in the IPTW method in more detail. In the IPTW method, we first estimate the treatment probabilities to construct the weights. The treatment probability (also called 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 propensity score for each unit $i$ as a function of a set of $K$-dimensional observed covariates $\boldsymbol{X_i}$, which impact host $i$'s decision of adoption of the algorithm (or assignment into the treatment condition). These covariates would also include all observed confounders that impact the adoption decision and are also correlated with the DV in the subsequent DiD regressions:

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

The observed covariates, $\boldsymbol{X_i}$, includes both time invariant as well as time varying observables that can impact adoption of the algorithm. Regarding the time varying observables, the caveat is that we can only take their pre-treatment values in the regression. Therefore, to capture the impact of the time varying observables in our analysis, we followed the prior literature by taking their pre-treatment values when estimating the adoption probabilities (Austin 2011) – that is, when estimating the adoption probability for each property, we took the values of the time varying observables at the beginning of Nov 2015, which was just before the algorithm was launched. We approximate propensity score $\widehat{ps}_i$ by estimating 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, which equals 1 if unit $i$ was observed to receive treatment and equals 0 if otherwise. The estimation process finds a parameter vector $\boldsymbol{\beta}$ that maximizes the data likelihood of observed treatment assignments (Rosenbaum and Rubin 1983), where $f$ (.) takes a logit functional form. Given the estimate of $\boldsymbol{\beta}$, we compute each unit $i$'s propensity score as $\widehat{ps}_i = f(\boldsymbol{X_i}\boldsymbol{\beta})$, which we use to compute the weights of the sub-sample of the observations of each property $i$ as ($T$ is the binary treatment indicator, which takes the value of 1 if the unit $i$ belongs to the treatment group)

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

These weights capture the contribution of different units when estimating the average treatment effects in the subsequent DiD regressions. In other words, we estimate the average treatment effects in the subsequent DiD regressions using weighted least squares, with $w_i$ being the weight of each unit *i*. By weighting each unit *i*'s observations by $w_i$, IPTW creates a 'synthetic sample' of weighted treatment and control groups in which the covariates X are balanced across the two weighted groups. Since the covariates X also include the potential observed confounders, the covariate balance across the weighted treatment and control groups ensures that the treatment assignment across the two weighted groups is independent of any observed confounder; further if there were no unobserved confounders, this would in turn imply that the weighted treatment and control groups are as good as data to being sampled from a population in which treatment assignments were random. This random treatment assignment would then ensure that the estimate of the average treatment effect is unbiased.

The rest of this section is organized as follows. In section 3.1, we list the covariates X that we use for estimating the propensity scores, along with the estimates of their coefficients. Following that in section 3.2, we do the covariate balance assessment to check whether the weighted treatment and control groups are balanced in terms of the covariates X. Finally in section 3.3, we assess the sensitivity of the estimates of treatment effects to unobserved confounders.

### 3.1. Estimation of the Treatment Probability

To improve the performance of IPTW and to reduce potential bias due to omitted variables in the subsequent DiD, we used a broad list of observed covariates, X, that are available to us for estimating the probability of treatment assignment of each property (i.e., the probability of adoption of the algorithm). These variables include the ones that that we discussed in section 2.4 of the Web Appendix. We estimate the probability as a logit function of the observed covariates X. In Table S3 we present the list of covariates used in the estimation of the adoption probability and the estimation results.

**Table S3 List of variables and estimation results in Treatment Probability Estimation.**

| **VARIABLES** | **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 | | |

### 3.2. Validation checks for IPTW: covariates balance assessments

Given the estimates, $\boldsymbol{\beta}$, we compute the weights of the sub-sample of the observations of each property *i* from the expression given in equation (S3). By weighting each unit *i*'s observations by $w_i$, the objective of the IPTW method is to create a synthetic sample of weighted treatment and control groups in which the covariates X are balanced across the two weighted groups. Thus our next step is to validate whether the resulting adopters and non-adopters (in the weighted sample) are comparable over the observed covariates that we listed in section 3.1. We thus assess the covariates balance on the weighted sample. Following prior literature (Rubin 2001, Stuart 2010), we compute the standardized difference in the means of the covariates across the two groups. This method compares, over *M*-dimensional covariates, the means of the treated group, $\overline{\boldsymbol{X}}_{\boldsymbol{treatment}}$, with the means of the control group, $\overline{\boldsymbol{X}}_{\boldsymbol{control}}$. The standardized differences are computed by normalizing the difference by the sample variance, $s^2_{treatment}$ and $s^2_{control}$:

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

We calculate the above statistic for each observed variable in the two weighted samples. The statistic $d^m$ stands for the standardized difference between the treatment and the control group along variable *m*. If $d^m$ is small, then it means the differences in that observed variable between the two weighted samples are small, which implies that the two groups are comparable in that observed variable. In the above equation, the means and variances in the treatment and control groups are weighted by the sample weight, represented by $\omega_i$. The means and variance for treatment and for control groups are:

$$\begin{cases} \overline{\boldsymbol{X}}_{\boldsymbol{treatment}} = \frac{\sum_{i \epsilon treatment} \omega_i \boldsymbol{X_i}}{\sum_{i \epsilon treatment} \omega_i} \\ s^2_{treatment} = \frac{\sum_i \omega_i}{(\sum_i \omega_i)^2 - \sum_i (\omega_i)^2} \sum_i \omega_i (X^m_i - \bar{X}^m_{treatment})^2 \end{cases} \quad i \; in \; treatment \; group \tag{S6}$$

$$\begin{cases} \overline{\boldsymbol{X}}_{\boldsymbol{control}} = \frac{\sum_{i \epsilon control} \omega_i \boldsymbol{X_i}}{\sum_{i \epsilon control} \omega_i} \\ s^2_{control} = \frac{\sum_i \omega_i}{(\sum_i \omega_i)^2 - \sum_i (\omega_i)^2} \sum_i \omega_i (X^m_i - \bar{X}^m_{control})^2 \end{cases} \quad i \; in \; control \; group \tag{S7}$$

where $\omega_i$ is the sample weight computed for unit *i* as inverse of the probability of treatment that unit *i* received. Specifically for unit i in the treatment group, $\omega_i = \frac{1}{\widehat{ps_i}}$, and for unit in the control group, $\omega_i = \frac{1}{1-\widehat{ps_i}}$ ($\widehat{ps_i}$is the estimated propensity of adopting smart pricing algorithm).

We next assess the sample balance by computing the standardized differences in the covariates. If for a given covariate, the absolute standardized difference between the control group and the treatment group is below 10% (i.e., 0.1), then it is considered that the imbalance is negligible (see discussion in Austin and Stuart 2015). As shown in Table S4, the absolute value of the standardized differences in all covariates (in the weighted sample) are well below 0.1. This indicates that our IPTW strategy created a weighted sample that effectively removed significant imbalances in the observed covariates that may have existed in the raw sample.

**Table S4 Validation test for IPTW: covariates balance check. Shows the standardized difference between the adopters and the non-adopters along the list of observed covariates. The statistics before IPTW-weighted (raw sample) and after IPTW-weighted (weighted sample) are both presented.**

| | Standardized Differences | |
|---|---|---|
| **VARIABLES** | **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 |

Note: the means and standard deviations of the covariates were computed for the pre-treatment (i.e., at the beginning of Nov. 2015) measurements.

### 3.3. Sensitivity analysis of Selection on Unobservables: Conditional c-Dependence Test

The unbiasedness of the treatment effect rests on the conditional independence assumption (CIA) that we have used in our IPTW and DiD analysis. The CIA assumption is that conditional on the distributions of the observed covariates *f(X)*, the potential outcome *Y* is independent of the treatment assignment *T*. Though the weighted sample of adopter units and non-adopter units are comparable along the list observed covariates, it is still possible that there were relevant variables omitted from the IPTW implementation. 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. Researchers have typically used Rosenbaum bounds sensitivity analysis to do that. In our study, we do not use the Rosenbaum sensitivity analysis. Instead we use the conditional c-dependence sensitivity test, which was recently proposed in the literature by Masten and Poirier (2018, 2019 and 2020) to assess the robustness of the average treatment effect (average impact of adoption on revenues) as well as the differential effect (differential impact of adoption on revenues of black vs. white hosts) to unobserved confounders. The conditional c-dependence sensitivity analysis does not impose parametric assumption on the modeling of the selection process or on the impact of treatment on the outcome variable. There are two reasons why we do not use the Rosenbaum bounds sensitivity analysis, and instead use the conditional c-dependence analysis. First, Rosenbaum sensitivity analysis cannot be employed when using IPTW. It can only be employed when we use PSM to create a matched sample. Second, the Rosenbaum bounds analysis cannot be used to test the robustness of the differential effect. It can only be used to test the robustness of the average treatment effect.

#### 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 of the conditional c-dependence exercise is to inform us how large the impact of hypothetical unobserved confounders need to be in order for the treatment effect to be nullified – the larger the value of unobserved confounders that is needed to nullify the treatment effect, the greater will be the confidence that our estimated treatment effect will be robust to unobserved confounders.

To see how conditional c-dependence analysis works, first note that in our IPTW and DiD analysis, we estimated the treatment effects based on the assumption that the conditional independence assumption (CIA) holds true. In other words, in our IPTW and DiD analysis, we estimated our treatment effects based on the assumption 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 X only. Note that this assumption holds true as long as there are no unobserved confounders. In contrast, Masten and Poirier (2018) proposed a class of assumptions that are weaker than CIA, namely the conditional c-dependence or partial independence, which can be used to assess the sensitivity of the treatment effects (obtained under CIA) to hypothetical unobserved confounders.

The conditional c-dependence assumption states that under a hypothetical scenario in which CIA is violated, the two aforementioned conditional probabilities would not be identical, but should be within a distance from each other. Under conditional c-dependence, the deviation from CIA, as captured by the maximum distance between the two aforementioned conditional probabilities, is described by a scalar variable $c$. If CIA were true, the two conditional probabilities would be equal to each other, and the value of $c$ will be zero. If CIA is violated, there will be a non-zero distance between the two conditional probabilities, bounded by the value of $c$. This distance $c$ captures the impact of unobserved confounders. The greater the magnitude of $c$, the greater will be the impact of unobserved confounders, and consequently the greater will be violation of CIA.

Given these primitives, the objective of the conditional $c$-dependence exercise is to calculate the minimum amount of deviation required from the CIA (or the minimum value of $c$) that would nullify the treatment effect that we had estimated assuming CIA to hold true. Intuitively, if it requires a large value of $c$ to nullify a treatment effect obtained under CIA (i.e., the hypothetical unobserved factors need to be so large enough to violate CIA), then it would imply that the estimate of the treatment effect that we had obtained by assuming CIA is quite robust to hypothetical unobserved confounders. On the other hand, if a small value of $c$ can nullify the treatment effect, then it would suggest that the estimate of the treatment effect that we had obtained under CIA is sensitive to the hypothetical unobserved confounders. This minimum value of $c$ required to nullify the treatment effect is called 'breakdown value'.

We next briefly discuss how the breakdown point is calculated. Formally, 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 for being selected to treatment (i.e., the property adopted the algorithm) and *T*=0 for not exposed to treatment (i.e., the property did not adopt the algorithm)
- *X*: the set of observed covariates used in IPTW
- *Y*: the observed outcome

Hence, we observe (*Y*,*T*,*X*), where only the $Y = \mathbf{1}\{T = 1\}Y_1 + \mathbf{1}\{T = 0\}Y_0$ is observed as the realized outcome. Under 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/adoption probability, $y_T \in \text{support of } (Y_T|X = x)$ and $x \in$ support of $(W)$. On the other hand, under conditional c-dependence, the independence only partially holds true, such that:

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

where sup-norm distance denotes how much the two adoption/treatment probabilities differ from each other. The scalar $c$ has a value between 0 and 1, where 0 refers to the full conditional independence assumption as a special case. The treatment effect based on IPTW and DiD analyses is point estimated when CIA holds true (i.e., when $c$=0 or when there are no unobserved confounders). On the other hand, Masten and Poirier (2018) show that if CIA is violated (i.e., when $c$>0 or when there are unobserved confounders), then the treatment effect based on IPTW and DiD analyses will not be point estimated and will instead be defined in a range for a given significance level. There will be an upper and lower bound around the estimated treatment effect, which 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 around the estimated treatment effect 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.

Given the upper and lower bounds vary as functions of $c$, we can then identify the breakdown point, $c_b$. The break down point, $c_b$, is defined as the minimum value of $c$ at which the data disproves the conjecture about the treatment at 95% significance level. When our conjectured value of the treatment effect is positive, the breakdown value, $c_b$, will be the value of $c$ when the lower bound curve intersects the baseline of zero (x axis); and when our conjectured value of the treatment effect is negative, the breakdown value, $c_b$, will be the value of c when 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 average impact of adoption on revenues) and the differential effect (the differential impact of adoption on revenues of black vs. white hosts). Thus in what follows, we report the results on $c_b$ for the average treatment effect in section 3.3.2. In section 3.3.3, we report the results on $c_b$ for the differential effect. And finally in section 3.3.4, we discuss what obtained values of $c_b$ imply in terms of robustness of the main and differential effects.

### 3.3.2. Results on conditional c-dependence sensitivity analysis: the average treatment effect

We present results on the upper and lower bounds and the corresponding breakdown point based on 95% significance level for the average treatment effect. For our implementation, we used Stata package – *tesensitivity* – developed by the authors of conditional c-dependence (Masten and Poirier 2019). This package is implemented following Stata's command – *teffects* – for estimating treatment effect using IPTW[11].

As shown in Table S5, the upper bound and the lower bound have the same value when $c$=0. That is, the treatment effect is point identified under CIA. Recall that that the size of $c$ depends on the size of the impact of the hypothetical unobservable. As the impact of the hypothetical unobservable increases (i.e., value of $c$ increases), we become more uncertain about the size of the treatment effect and can only place bounds (as opposed to point estimates) on the estimate (these are bounds around the estimate of the average treatment effect). The formula that the software used

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

to compute the bounds as well as their theoretical derivations and proof are in Theorem 1 (Proposition 3-Corllary 1 and Proposition 4 of Masten and Poirier 2018).

To see this graphically in Figure S2, the bounds become wider as value c increases. Here we have shown the results for the first 10 values of $c$. As shown in Figure S2 and Table S5 the breakdown point is $c_b = 0.083$ for the average treatment effect, suggesting that the closed interval on the estimated treatment effect would start to include 0, above the value of 0.083[12]. The estimated breakdown point implies that in order for our estimated average treatment effect of the pricing algorithm to be nullified, the potential unobservables that enter the adoption decision of the algorithm must be large enough so that the unobservables cause a deviation of at least 0.083 in the adoption probability..

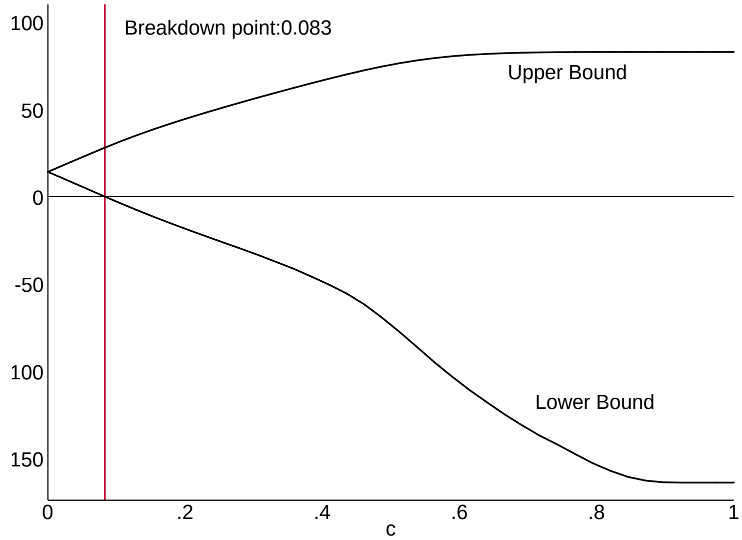
In section 3.3.4, we will benchmark the breakdown point in terms of the results obtained in prior studies that have used conditional c-dependence and other sensitivity analyses. That will give a better picture of how robust our results are to hypothetical unobserved confounders.

**Table S5 Treatment effects sensitivity: *conditional c-dependence b*ounds**

| **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 |
| Outcome mode: logistic<br>Treatment model: linear quantile<br>Outcome variable: Average Daily Revenue over Adoption Periods<br>Conjecture: β> 0<br>Breakdown point: 0.083 | | |

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

**Figure S2 The *Conditional c-Dependence* Bounds on Effect of Using Smart Pricing Algorithm on the Airbnb Property *Revenue*: Bounds vary over the value of c**




**3.3.3. Results on conditional c-dependence sensitivity analysis: the differential effect[13]**

In this section we assess how sensitive the differential effect is to any violation of CIA. To do so, we first conducted the conditional c-dependence analysis separately for two subgroups in the sample—for hosts in the white and the black ethnicity groups, respectively. Then we examined the value of $c$ when the bounds on the estimated treatment effects for the two subgroups start to overlap. That value of c would be the breakdown point of the estimate of differential effect. Essentially, we first computed the average treatment effect conditional on each ethnicity group, then computed the bounds for each treatment effect, when in turn informed us on the value of c when the bounds of the two groups start to overlap.

To see this formally, here we are interested in the difference between the treatment effects for white and for black ethnic groups, i.e.,

$$\Delta\beta = \beta_{black} - \beta_{white} \tag{S10}$$

where $\beta_{black}$ and $\beta_{white}$ are the treatment effect of using smart pricing on the property demand for black hosts, and for white hosts, respectively. We can construct bounds on the difference as:

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

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

where $UB$ and $LB$ refers to upper bound, and lower bound, respectively. Then we could just plot that single set of bounds, ($UB_{\beta_{black}}$, $LB_{\beta_{black}}$, $UB_{\beta_{white}}$, $LB_{\beta_{white}}$), as functions of $c$. Note that $LB_{\beta_{black}}$ is the lower bound of the estimated treatment effect for the black ethnic group and $UB_{\beta_{white}}$ is the upper bound of the estimated treatment effect for the white ethnic group, in the presence of uncertainty. The value $c$ where $LB_{\beta_{black}}$ and $UB_{\beta_{white}}$ first intersect (as we are interested in testing $\beta_{black} > \beta_{white}$) 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. Note that the with way we compute the bounds for the differential effect, we should interpret the sensitivity results as conservative estimates. This is because looking at the lower bound of black

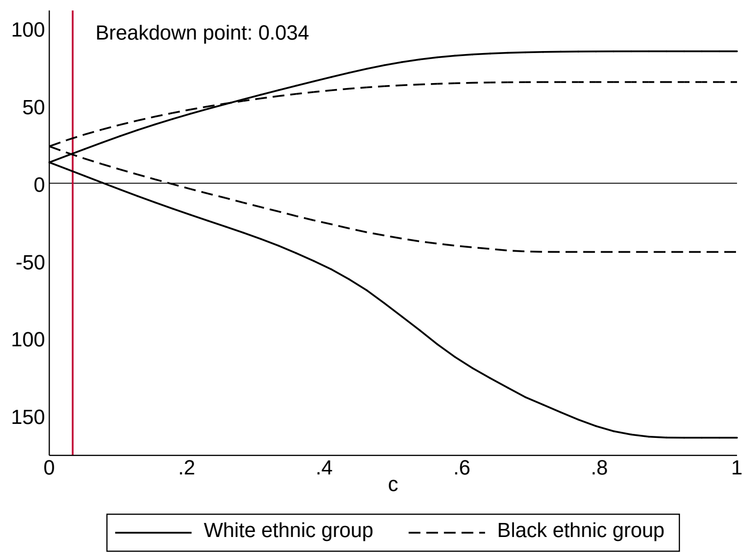
[13] We thank Matthew Masten (author of Masten and Poirier 2018, 2019) for suggesting the methodology for calculating the breakdown value of the differential effect.

($LB_{\beta_{black}}$) and upper bound of white ($UB_{\beta_{white}}$) is a conservative comparison, in the sense that it is assuming the data lines up to make the difference $\Delta\beta = \beta_{black} - \beta_{white}$ as small as possible.

In

Figure **S3** we plot the bounds for the estimated treatment effect for white, and black, ethnic groups. Clearly, the difference (or amount that $\beta_{black}$ above $\beta_{white}$) diminishes as $c$ increases. The value of the breakdown point $c_b$ is 0.034 – it is the point above which the two intervals on the estimated treatment effects for white and black ethnic groups would start to overlap. This breakdown point implies that in order for our estimated differential effect of the pricing algorithm to be nullified, the potential unobservables that enter the adoption decision of the algorithm must be large enough so that the unobservables cause a deviation of at least 0.034 in the adoption probability. In section 3.3.4, we will benchmark the breakdown point value in terms of the results obtained in prior studies that have used conditional c dependence and other sensitivity analyses. That will give a clearer picture of how robust our results are to hypothetical unobserved confounders.

**Figure S3 Conditional c-Dependence Bounds on Effect of Using Smart Pricing Algorithm on the Airbnb Property Revenue: For White and Black Ethnic Group Separately**




### 3.3.4. Interpretation of conditional c-dependence sensitivity analyses

Recall that our estimate of breakdown point is 0.083 for the average treatment effect, and 0.034 for the differential effect. To understand what these values imply in terms of percentage changes in adoption probabilities, note that the mean treatment (adoption) probability in our sample is 19.5%. Thus if we were translate our result in terms of percentage changes in adoption probabilities, our estimated breakdown points imply that in order for our estimated average treatment (differential) effect of the pricing algorithm to be nullified, the potential unobservables that enter the adoption decision of the algorithm must be large enough so that the unobservables cause a deviation of at least 0.083/0.195=42.6% (0.034/0.195 =17.4%) change in the adoption probability.

Next, to get a better sense of what these breakdown values imply in terms of robustness, we benchmarked these in two ways. First, we will benchmark our results with the robustness measures obtained in prior studies that have used conditional c-dependence analysis. Following that, we will benchmark our results with the robustness measures obtained in prior studies that have used the Rosenbaum bounds analysis. As discussed earlier, Rosenbaum bounds analysis a widely-used sensitivity analysis when using PSM.

1. *Comparison with Robustness Measures obtained in Prior Studies that have used Conditional c-Dependence Analysis:* We first benchmarked our results with the breakdown values obtained in the prior literature that has used conditional c-dependence analysis. Since conditional c - dependence analysis is a recent development, there are only a few applications. Masten and Poirier (2018, 2019) performed the conditional c-dependence analysis on the well known *Lalonde1986* dataset that has been used by researchers to measure the causal effect of education on income. This dataset consists of 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). Masten and Poirier (2019) performed the conditional c-dependence analysis on these two samples, and obtained the breakdown values of the treatment effect as 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 the values of the breakdown points that we obtain in our analysis are larger than 0.020, it suggests that our estimates of both the average treatment and the differential effects are fairly robust to unobserved confounders. Recall that when analyzing the sensitivity for the differential effect, we assumed that the data lines up to make the additional increase for black host, compared to the effect for white hosts, as small as possible. Hence, in the way we compared the bounds for the treatment effect for black versus for white ethnic groups, the sensitivity result we obtained for the differential effect is a conservative estimate.

*2. Comparison with Robustness Measures obtained in Prior Studies that have used Rosenbaum Bounds Analysis:*[14] 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). Similar to conditional c-dependence analysis, Rosenbaun bounds analysis identifies the minimum value of the impact of unobservables that enter the treatment selection process at which the treatment effect is nullified. However unlike the conditional c-dependence analysis in which the impact of unobservables is captured in terms of the absolute distance between the conditional probabilities (as in equation S9), the impact of unobservables in Rosenbaum bounds analysis is captured in terms of change in

[14] A caveat is in order here since we are comparing robustness measures across two different methodologies. The reason why we have made this comparison is because there are very few studies that have used the conditional c-dependence analysis, and there are lot more studies have used Rosenbaum bounds analysis. Thus comparing our robustness measures with those obtained in the past studies that have used Rosenbaum bounds analysis is the next best option. Based on our correspondence with Matthew Masten (author of Masten and Poirier 2018), the results of these two methodologies are similar, and they yield almost identical results when we have a one on one matched sample based on PSM.

odds ratio of being treated between two units that are otherwise similar on the observables. Thus unlike the conditional c-dependence test in which the robustness of the treatment effect is measured in terms of the breakdown value (which is the minimum value of the absolute distance between the conditional probabilities at which the treatment effect is nullified), the robustness of the treatment effect in the Rosenbaum bounds analysis is measured in terms of the Gamma value, which is the minimum value of the change in odds ratio at which the treatment effect is nullified.

The above discussion implies that if we were to compare the robustness measures across conditional c-dependence analysis and Rosenbaum bounds analysis, we would need to translate the breakdown value obtained from the conditional c-dependence analysis to 'the change in odds ratio'. This would be done as follows. Suppose there exist hypothetical unobservables that cause a deviation in the adoption probability by an amount of $c_b$—the breakdown point value. We then compute, for an average property, what the 'altered' adoption probability would be under the impact of hypothetical unobservables, and accordingly what the 'altered' odds would be. The change in the odds ratio would give us the equivalent Gamma value. In our sample, the adoption odds for the average property (given that the mean adoption probability in our sample is 19.5%) is 19.5%/(1-19.5%)) = 0.24. If the hypothetical unobservables caused a deviation in the adoption probability by $c_b$=0.083 for the average treatment effect, then the altered odds ratio would be (19.5%+8.3%)/[1-(19.5%+8.3%)] =0.39. Thus the change in the odds ratio for $c_b$=0.083 will be 0.39/0.24 =1.63, which is the equivalent Gamma value for the average treatment effect in the Rosenbaum bounds analysis. Similarly, the impact of $c_b$=0.034 for the differential effect can be translated in term of change in odds ratio to a Gamma value of 1.25.

This translation implies that in order to nullify our inference of the average treatment (differential) effect at 95% significance level, the potential unobserved confounders that enter the adoption decision must be large enough so that the Airbnb units in our sample must be 63% (25%) more likely in terms of odds-ratio to adopt the algorithm. Our values of Gamma for both the main and differential effects are similar to the ones in the prior literature that reported values from 1.2 to 1.6 (e.g., DiPrete et al. 2004, Sun and Zhu 2014, Manchanda et al. 2015). This once again implies that our results are fairly robust to potential unobserved confounders.

*3. Interpreting the results using one-leave-out analysis*: We have present additional analyses that allows us to interpret the breakdown values of the average treatment effect and the differential effect in terms of the impact of a known covariate on the adoption probability. The additional analysis is called the 'leave-one-out' analysis (Masten and Poirier 2018, 2020), which we explain as follows.

Recall that for the average treatment effect (differential effect), the estimate of the breakdown value was $c$=0.083 (0.034), which implies that the hypothetical unobservable needs to cause a deviation of at least 8.3% (3.4%) in order to nullify the average treatment effect (differential effect). In this regard, an alternative to interpret the value of $c$ is to identify the covariates that would cause a shift in the adoption probability by a similar magnitude that our obtained breakdown values would imply (which is 8.3% for the average treatment effect and 3.4% for the differential effect). The leave one out analysis helps with this task. In the leave-one-out analysis, for each given covariate $X^m$ in the adoption model, we compare the following two estimated adoption probabilities: (i) the estimated probability obtained when we include all the covariates that we had used in the adoption probability model, and (ii) the estimated adoption probability obtained when we leave out the covariate $X^m$ from the adoption model, and (ii). Comparing these two probabilities for each covariate $X^m \in \{X^1..X^M\}$, will allow us to assess the impact of each of the covariates on the adoption probability. Following that, we can then assess which of the *M* covariates have an impact greater than what the breakdown values of the average treatment effect would imply (which is 8.3%) and the differential effect would imply (which is 3.4%), and also identify the specific covariates whose impact is the same as what the breakdown values of the treatment effect and the differential effect would imply.

In more specific terms, suppose in our main analysis we used all covariates $\{X^m\}_{m=1}^{M}$ and computed the adoption probability for each property *i* as a function of the covariates: $p_i = f(X_i^1, X_i^2, \ldots, X_i^M)$. The leave-one-out analysis then computes *M* sets of alternative adoption probabilities, where for each $m^{th}$ covariate excluded from the estimation, it estimates $p_i^{-m} = f(X_i^1, X_i^2, \ldots, X_i^{m-1}, X_i^{m=1}, \ldots, X_i^M)$. Here -*m* indicates that covariate $X^m$ was not used in estimating the adoption probability. Comparing the two adoption probabilities $p_i$ and $p_i^{-m}$ will tell us the deviation in the adoption probability for individual *i* when we exclude $X^m$. For a given property i, this deviation will be a point value. However, across the population of properties, the deviation in the adoption probability when we exclude the covariate $X^m$ will be a distribution, since the value of $X^m$ is itself randomly distributed across properties in the sample. To run the analysis, we have used 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 have reported the results in Table S6 below. To assess the impact of omitting each variable $X^m \epsilon \{X^1..X^M\}$, we typically look at the $50^{th}$ percentile in the distribution of its deviation in the adoption probability. Given these $50^{th}$ percentile values across all covariates, we then search for the covariate for which the $50^{th}$ percentile in its distribution of the deviation in adoption probability is greater than 0.083 (for the average treatment effect) and greater than 0.034 (for the differential effect). Looking at the $50^{th}$ percentiles (column '0.5 (median)') across all covariates in Table S6, we see that none of covariates are associated with a deviation in the adoption probability that is as large as 0.083. Even when we look at the maximum possible deviation in the adoption probability caused by each 'omitted' covariate (column 'max'), only 13 of the 44 covariates achieve a deviation comparable to 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*. Out of these 13 covariates, the breakdown value of 0.083 is closest to the maximum deviation that # Bedrooms would cause on the propensity score (=0.082). Hence, the estimated breakdown value of 0.083 for the average treatment effect suggests that in order for the average treatment effect to be nullified, the impact of hypothetical unobserved confounders on the adoption probability needs to close to the maximum impact that the variable, size of the property (in terms of # Bedrooms), has been observed to have in our sample.

We next do the same for the differential effect. Looking at the $50^{th}$ percentiles (column '0.5 (median)') across all covariates in 10, we see that none of covariates are associated with a deviation in propensity scores that is as large as 0.034. If we compare the maximum deviation in propensity score (see column 'max' in Table S6), the breakdown value is still *greater* than the maximum impact of the following 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*. Out of these 14 covariates, the breakdown value of 0.034 is closest to the maximum deviation that # Reservation Days would cause on the propensity score (=0.036). Hence, the estimated breakdown value of 0.034 for the differential effect suggests that in order for the differential effect to be nullified, the impact of hypothetical unobserved confounders on the adoption probability needs to close to the maximum impact that the variable, # Reservation days, was observed to have in our sample.

**Table S6 Interpreting Sensitivity Analysis: Leave-one-out on the Difference in Propensity Scores**

| Analysis: leave one out propensity score difference | | | | |
|---|---|---|---|---|
| **'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 |
| Treatment model: logistic regression | | | | |

## Section 4. DiD Regressions: Analysis and Variables

We implement Difference-in-Difference analyses on the constructed weighted sample to estimate the effect of using the smart pricing algorithm on revenues, prices and occupancy rate. That is, estimating the DiD model on our sample, with the computed sample weights enter the estimation in a WLS manner. In our study, we include property fixed effects, city specific monthly fixed effects and city specific year effects. We model the outcome variable (which is either average daily revenue or average daily price or occupancy), for property *i* in period *t* as follows:

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

where $\varepsilon_{it}$ is idiosyncratic shock in $Y_{it}$. Property fixed effects, $Property_i$, capture time-invariant factors and control for the time-invariant confounders, such as property size, location, hosts' demographics, local demographic, unobserved property quality. Seasonality fixed effects, $Seasonality_t$, includes the city-specific yearly and monthly fixed effects that impact Airbnb property's revenue. As discussed in the paper, since the primary factors that impact the adoption decision are time invariant, incorporating the property fixed effects would control for all the primary confounding factors. $Controls_{it}$ represent a set of covariates that may correlate with $Y_{it}$. The variable $SmartPricing_{it}$ equals 1(0) if the prices in period $t$ of property $i$ were (not) determined by the pricing algorithm. Thus, the DiD estimator—$\beta$—identifies the change in the economic outcomes caused by the pricing algorithm, i.e., the effect of pricing algorithm on $Y_{it}$.

**List of variables in DiD**

In our DiD specification as shown in Equation (S7), the property fixed effect $Property_i$ takes care of all time invariant unobserved confounders. For the variables that are time-varying, we include them in $Controls_{it}$. 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 List of control variables used in DiD. These variables may vary over time and correlate with property's daily revenue. Thse statistics are presented by adoption group.**

| VARIABLES | (1) Adopters | | (2) Non-adopters | | (3) All Properties | |
|---|---|---|---|---|---|---|
| | 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 |

**Section 5. Second Stage Regression for Estimating the Main Effect of the variable 'Black'**

In this section, we explain how we computed the main effect of the variable 'Black' in the DiD regression given in equation (2) in the main paper which we re-write as follows.

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

In this regression the DV is $Y_{it}$, which can take the value of either average daily revenue per month, average daily price per month or average occupancy per month. Note that the main effect of Black cannot be directly identified from the regression in equation (S14) because it is absorbed in the property fixed effects $Property_i$. Thus we first ran the regression in equation (S14) from which we obtained the property fixed effect, $Property_i$, for each property $i$. The estimate, $Property_i$ then captures the time-invariant factors, for example, neighborhood, host demographic, that impact daily revenue for $i$ prior to the launch of smart pricing. Then in the second step, we regress all estimated $Property_i$ on the timeinvariant property and neighborhood characteristics, as shown in Equation (S15) using minimum distance estimation (as discussed in Nevo 2000 and Chamberlain 1982):

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

where we include host's ethnicity and let *White* serve as the reference ethnicity group. The purpose is to estimate the coefficient of key variable, *Black*, which captures the impact of being a black host, compared to a white host, on property's daily revenue. The coefficient $\alpha$ with respect to *Black* represents the main effect of the variable, Black. For the case when the DV in equation (S14) is revenues, $\alpha$ will represent the amount of revenue that black hosts earned over and above the white hosts prior to the adoption of the algorithm, conditional on all property fixed effects and all other host and property characteristics. Similarly, for the case when the DV in equation (S14) is average nightly rate (occupancy), $\alpha$ will represent the price (occupancy) difference between black and white hosts prior to the adoption of the algorithm, conditional on all property fixed effects and all other host and property 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 make a proper comparison between black and white hosts, we need to take into account the differences (in the host, property and neighborhood characteristics) between the two. Hence in Equation (S15) we incorporate all the observed property, host characteristics that are available to us. $Characteristics_i$ represents a set of covariates that may correlate with $Property_i$. See Table S8 for the list of these covariates. We also include neighborhood fixed effects, $Neighborhood_n$, to capture unobserved time-invariant neighborhood factors that may affect $Property_i$ (such unobserved factors would include location convenience, traffic, nearby hotels/restaurants/malls, and local demographics within the neighborhood). 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 a DV in the second stage regression in which we regress it on time invariant property and host characteristics, we need to account for the fact that has sampling error

– otherwise we will under-estimate the standard errors of the estimates in the second stage regression. To do so, we followed Nevo (2000, 2001) and used Minimum-Distance Estimation in the 2nd stage regression. In his work, Nevo (2001) first 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) using Minimum-Distance Estimation proposed by Chamberlain (1982).

Specifically, following Nevo (2001), if we let $f$ as the $J \times 1$ property fixed effects that we obtained from the first-stage regression and $X$ be the $J \times K$ property attributes (e.g., the property characteristics, host ethnicities, neighborhood fixed effects etc.), then in the second-stage, we regress property fixed effects on X:

$$f \sim X\beta + \epsilon \tag{S16}$$

where $\epsilon$ is the error term. The estimate of coefficient for property and host characteristics can be written as:

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

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 the estimation of second-stage can be seen as a generalized least squares (GLS) regression where the correlation in the dependent $\hat{f}$ are weighted by the estimated covariance matrix $V_f$. The standard errors (variance matrix) were computed using standard formulas of the standard errors in a two-step parametric M-estimator (Hansen 1982, Newey and McFadden 1994)). Specifically, if we specify the two stage coefficients as $\hat{\alpha} = \left(\hat{f}, \hat{\beta}\right)'$, then with $\alpha^*$ indicate the true values, the asymptotic variance of $\sqrt{n}(\hat{\alpha} - \alpha^*)$ is given by

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

where $g(\cdot)$ in consists the first order conditoins for the M-estimators in the two step of regressions.

The estimates of the second stage regression related to revenue are reported in the first column of Table S7. Observe that the coefficient of key variable, *Black*, is negative and significant ($\alpha$=-12.16, $p$<0.001). The coefficient implies that there was a gap of $12.16 in the average daily revenues earned between white and black hosts (conditional on all other observed property and hosts characteristics). The estimates of the second stage regression related to average nightly rate are reported in the second column of Table S7. Observe that the coefficient of key variable, *Black*, is insignificant, which implies that both black and white hosts were charging similar prices prior to the introduction of the algorithm (conditional on all other observed property and hosts characteristics). The estimates of the second stage regression related to occupancy are reported in the third column of Table S7. Observe that the coefficient of key variable, *Black*, is negative and significant ($\alpha$=-0.104, $p$<0.001). The coefficient implies that the occupancy of black hosts was 0.104 lower than that of 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: Shows the effect of black ethnicity, compared to white ethnicity, on host's average daily revenue, average nightly rate, and average occupancy rate, prior to the introduction of smart pricing algorithm. The dependent variables are property-specific fixed effects. Hence the number of observations equals the number of properties in our sample. Base ethnicity group: white.**

| | **(1) Gap in Daily Revenue** | | **(2) Gap in Nightly Rate** | | **(3) Gap in Occupancy Rate** | |
|---|---|---|---|---|---|---|
| **VARIABLES** | **Coefficient** | **Std. Err.** | **Coefficient** | **Std. Err.** | **Coefficient** | **Std. Err.** |
| ***Black*** | **-12.16*** ** | **(2.095)** | **2.024** | **(4.769)** | **-0.104*** ** | **(0.0104)** |
| *Others* | -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 | 9396 | | 9396 | | 9396 | |
| R-squared | 0.31 | | 0.56 | | 0.24 | |

Note: only pre-treatment period (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.
D.V. is the individual property fixed effect that we estimated from the first-step DiD regression, where in the first-stage regression the depend variable was daily revenue (column 1), nightly rate (column 2), and occupancy rate (column 3).
Cluster-robust standard errors at individual neighborhood level in parentheses.
* p<0.05 ** p<0.01 *** p<0.001

## Section 6. Robustness Checks

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

We examine the pre-treatment trends to ensure that the weighted sample of adopters and non-adopters followed similar trends in their property revenues. The standard approach of doing so is by estimating a relative-time model (Autor 2003), which decomposes the pretreatment periods in terms of a series of period dummies and estimates the coefficients of all those dummies that are within *k* periods prior to the treatment. Specifically, we decompose the pre-treatment periods and examine the following dummies: *Pre(k)* indicates the $k^{th}$ period prior to the smart pricing algorithm adoption (for k=1,2,…5). *Pre(6)* represents the time segment from the beginning of our observational window all the way to the $6^{th}$ period prior to smart pricing algorithm adoption.

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

where $Adopter_i$ equals 1 if property *i* is in the treatment group—*i is* observed to adopt the pricing algorithm, and equals 0 if otherwise. The series of time dummies $Pre(j)_{it}$ equals 1 if period *t* is j periods prior to the period when the smart pricing adoption took place for property *i* and equals 0 if otherwise. The parameters $\{\alpha_j\}$ hence 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 reference period (i.e. $\{\alpha_j\}_{j=1}$ was normalized to zero). For a positive estimated treatment effect to be valid, we would expect that the set of $\alpha_j$ are not positive and significant.

Table S9 reports the estimation results for the IPTW weighted sample. As we can see, the estimated coefficients for the pre-treatment dummies are all statistically insignificant. Thus the parallel trends assumption is not rejected.

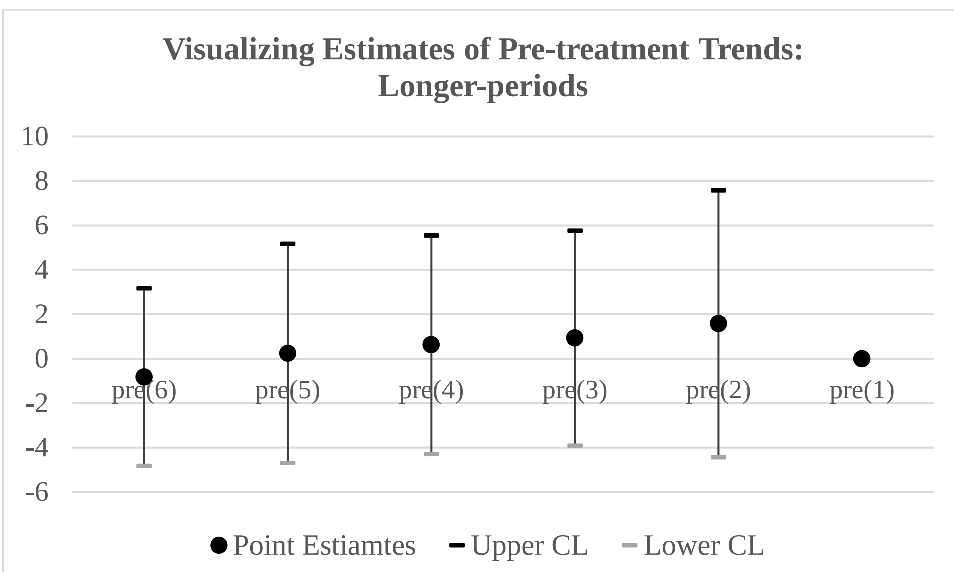
**Table S9 Validating DiD Model: Assessing Parallel Trends Assumption in Pre-treatment Periods for the IPTW weighted sample**

| VARIABLES | 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 Smart Pricing on 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 | |
| Cluster-robust standard errors at individual property level in parentheses<br>* p<0.05   ** p<0.01   *** p<0.001 | | |

***Visualizing Parallel Trends:*** In Figure S4 we plot the estimated coefficients that represent the pre-treatment trends for the IPTW sample. Note that the 95% confidence intervals of all coefficients in the pre-treatment periods contain zero.

**Figure S4 Plot of Estimated Coefficients in Pre-treatment Periods for the IPTW sample**



*Testing Pre-treatment Trends for DiD Analysis on PSM matched sample*

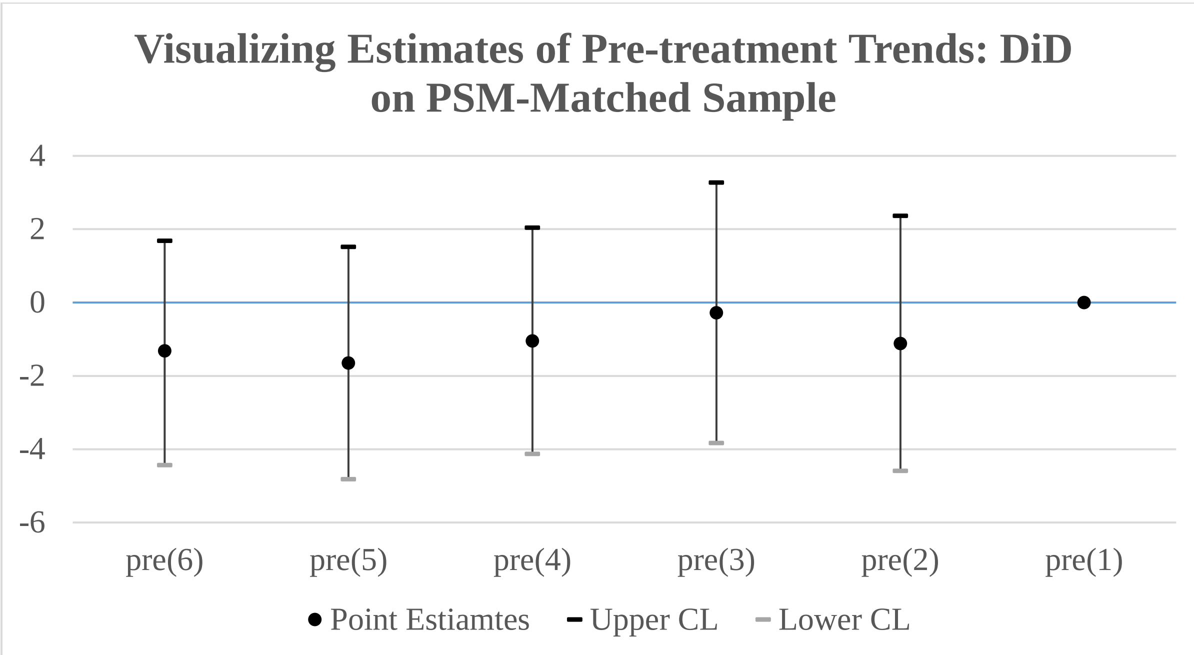
We examine the pre-treatment trends in the sample of adopters and non-adopters based on Propensity score matching (PSM). We use the same relative-time model that we used to assess the common pre-treatment trends assumption, as describe in this section. The only difference being that, we regress the relative-time model on the PSM sample.

The table below (Table S10) reports the estimation results. As we can see, none of the estimated coefficients for the pre-treatment dummies are statistically significant, suggesting that in the PSM sample, there do not exist significant difference in the outcome variables between the adopters and the non-adopters, prior to the treatment. In Figure S6 figure below we plot the estimated coefficients that represent the pre-treatment trends. Consistently, the 95% confidence intervals of all coefficients in the pre-treatment periods contain zero, showing common pre-treatment trends.

**Table S10 Assessing Pre-treatment Trends: DiD on PSM-Matched**

| **VARIABLES** | **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 Smart Pricing on 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 | |
| Cluster-robust standard errors at individual property level in parentheses<br>* p<0.05 ** p<0.01 *** p<0.001 | | |

**Figure S5 Plot of Estimated Coefficients in Pre-treatment Periods: PSM-Matched Sample**




***Testing Pre-treatment Trends for DiD on the Raw Sample (i.e., no matching/weighting)***

We examine the pre-treatment trends in the raw sample of adopters and non-adopters (that is, without any matching or weighting), in their property revenues. We use the same relative-time model that we used to assess the common pre-treatment trends assumption, as describe in this section. The only difference being that, we regress the relative-time model on the raw sample.

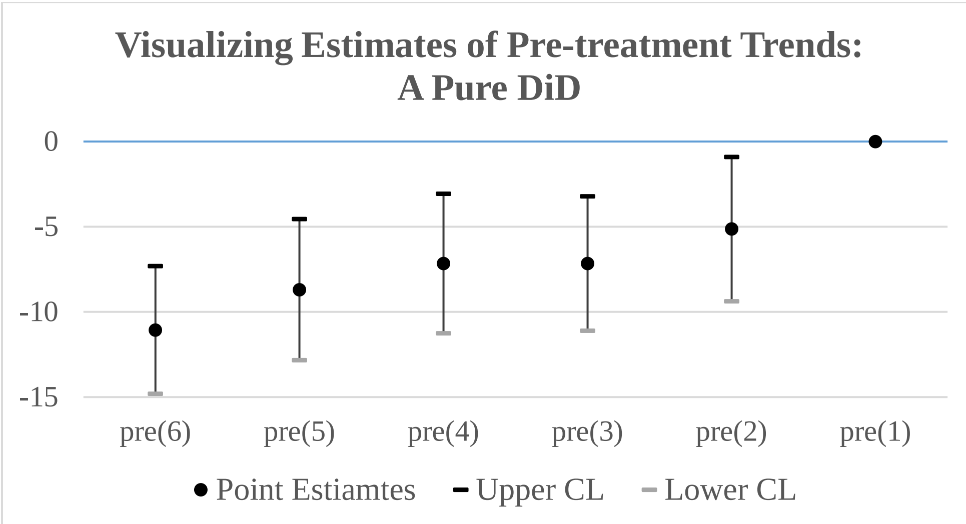
Table S11 reports the estimation results. As we can see, the estimated coefficients for the pre-treatment dummies are all statistically significant, suggesting that in the raw sample, there existed significant difference in the outcome variables between the adopters and the non-adopters, prior to the treatment. In Figure S6 we plot the estimated coefficients that represent the pre-treatment trends. The 95% confidence intervals of all coefficients in the pre-treatment periods do not contain zero, showing a lack of common pre-treatment trends.

Our results show that the parallel trends assumption for the pre-trends was rejected on the raw sample. This implies that in the raw sample, there were systematic differences across the treatment and control groups that can influence the outcome variable and the adoption decision. On the other hand, our results show that parallel trends assumption for the pre-trends was not rejected on the samples based on either PSM or IPTW. This suggests that PSM and IPTW do help in terms of identification to the extent that the systematic differences in the two groups that can influence the outcome variable are reduced (so that the parallel trends assumption is not rejected).

**Table S11 Assessing Pre-treatment Trends for the raw sample (without matching or weighting)**

| **VARIABLES** | **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 Smart Pricing on 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 | |
| Cluster-robust standard errors at individual property level in parentheses<br>* p<0.05 ** p<0.01 *** p<0.001 | | |

**Figure S6 Plot of Estimated Coefficients in Pre-treatment Periods: Lack of Parallel Trends when only DiD was performed on the raw (unmatched/unweighted) sample**




### 6.2 Validating DiD Model: Assessing Parallel Trends between Adopters and non-adopters in White and Black Ethnic Groups for IPTW weighted sample

We examine the pre-treatment trends for two ethnic groups of hosts, namely black hosts and white hosts for the IPTW weighted sample. Similar to Section 6.1, we examined the trends for a 6-period segment in the pre-treatment period. To assess the within ethnic group (i.e., white/black) trends, we interacted the ethnicity (white as reference ethnic group) of hosts with the series of pre-period dummies, and with the *Adopter* dummy (see section 6.1):

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

where is similar to what we did in section 6.1, we set the period prior to adoption month as the reference period, i.e., we normalize *Pre(1)* to zero. Since the reference ethnicity is white, the series of coefficients $\{\alpha_j\}_{j=2}^{6}$ capture the pre-treatment trend in the dependent variable within the white ethnic group, and the series of coefficients $\{\alpha_j + \eta_j\}_{j=2}^{6}$ in the series of the three-way interaction terms capture the pre-treatment trend in the dependent variable within the black ethnic group.

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. Note that $\alpha_j$ and $\eta_j$ were obtained from the same regression, in which the rest of the parameters have been pooled across hosts with different ethnicities.[15] As can be seen, the estimated coefficients for the pre-treatment dummies are all

[15] Note that the nature of the results does not change even if we were to estimate the other parameters separately for black and white hosts.

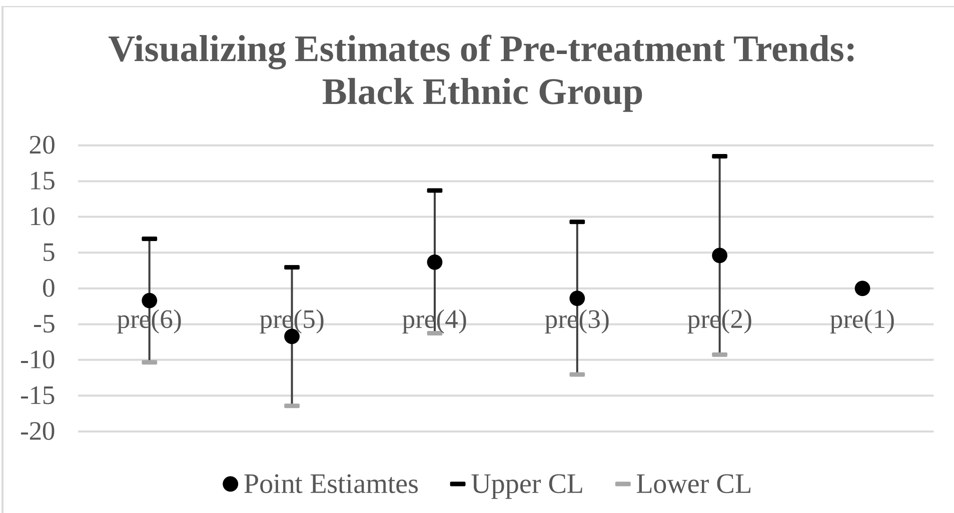
statistically insignificant for each separate ethnic group. In addition, non-of the series of time-trends exhibits a trend line that is increasing towards the adoption of smart pricing algorithm. Thus the parallel trends assumption for each of the two ethnic sub groups is not rejected. In Figure S7, we plot the pre-treatment trend for white, and for black, ethnic groups of hosts, respectively. As can be seen, within each ethnic group, for the adopters and non-adopters, their property daily revenues were not statistically different prior to their adoption of smart pricing algorithm.

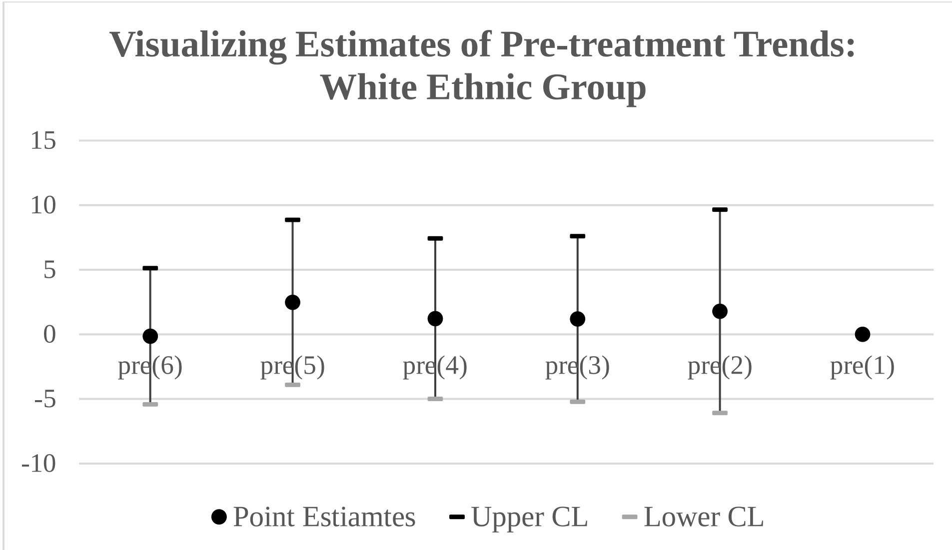
**Table S12 Validating DiD Model: Assessing Parallel Trends Assumption in Pre-treatment Periods with separate pre-treatment trends for Black and White Hosts for IPTW sample**

| **VARIABLES** | **ESTIMATES** | |
|---|---|---|
| | **White Group (reference)** | **Black Group** |
| **Pre-Treatment Trends (White: reference group)** | | |
| *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) |
| *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 |
| Note: 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. Cluster-robust standard errors at individual property level in parentheses * p<0.05 ** p<0.01 *** p<0.001 | |

**Figure S7 Plot of Estimated Coefficients in Pre-treatment Periods: Assessing Parallel Trends Assumption in Pre-treatment Periods, Separate Assessment by Host Ethnic Group for IPTW sample**






**6.3 Dynamic Treatment Effects**

We examine whether or not and to what extent there are dynamics in the treatment effect of the smart pricing algorithm on revenues. There are potentially three sources of dynamic treatment effects. First, since Airbnb may change the way its algorithm computes and recommends its prices to hosts over time, its impact on revenues would vary over time. Second, the impact of adoption of the algorithm on revenues of a host will diminish over time as the number of other properties that adopt the algorithm in the neighborhood increase. Third, the treatment effect could vary across peak and off-peak seasons.

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) will only lead to seasonal variations in the treatment effect.

It is easy to identify the dynamic impact of the third source since it only results in seasonal variations in the treatment effect. However, since we do not know when Airbnb made changes in its algorithm, and since we do not have access to a variable that can 'accurately' capture the competitive effect from other properties that have adopted the algorithm (more on this later), it is difficult to separately identify the dynamic impact of the first two sources. Nevertheless, we run two regressions to identify these dynamic treatment effects. In the first regression, we identify the dynamic effect of the third source (seasonality) and the joint effect from the first two sources. In the second regression, we will attempt to identify the dynamic effect of each of the three sources.

Starting with the first regression, in order to identify the dynamics from the third source and the joint effect of the first two sources, we regress revenue on three-way interaction terms involving the following variables:

(i) Adoption of the algorithm ($SmartPricing_t$), which is a dummy variable that takes a value of 1 if the focal host/property adopted the algorithm in the given time period;

(ii) Seasonality: For simplicity, we classify seasonality in terms of two seasons only: peak and off-peak seasons with peak season being the baseline season. For each city, we classified each month into either peak or off-peak season[16]. We represent this variable by a seasonal dummy, $OffPeak_t$, which takes the value of 1 if the month $t$ was in the off-peak season.

(iii) Number of years lapsed since the launch of the algorithm ($PostYear_t$): If the observation month $t$ in the data lies between Nov 2015 (which is the time period when the algorithm was launched) and Oct 2016, $PostYear_t$ takes a value of 1, and if the observation month $t$ in the data lies between Nov 2016 to Aug 2017, $PostYear_t$ a value of 2. Also $PostYear_t$ =1 includes pre-launch periods. This does not make any material difference since smart pricing adoption is zero for the pre-launch period.

In this regression, the interaction terms involving adoption of the algorithm with seasonality will capture the dynamics the in treatment effects that stem form the third source, and the interaction terms involving the adoption of the algorithm and number of years lapsed since the launch of the algorithm will capture the dynamics that stem from the first two sources combined.

The results of the regression are reported in column 1 of Table S13. The parameters of interest related to the joint effect of the first two sources are the coefficients of the interaction terms, (*PostYear=2)×SmartPricing* and (*PostYear=2)×SmartPricing×OffPeak*. Observe that the coefficients of these interaction terms are negative but non-significant. This suggests that there may not be significant dynamics in the treatment effects that stem from the joint effect of the first two sources.

In the next regression, we will try to separately identify the dynamics from all three sources. To separately identify the second source (the competitive effect) from the other sources, we introduce an additional variable in the above regression. This variable is Zip-code adoption rate, which is defined the fraction of listings in the same zip-code as that of the focal property that have adopted the algorithm at that point in time. The interaction of 'Zip code adoption rate' with the variable, 'adoption of the algorithm' will capture the impact of the second source. We thus regress revenue on four way interaction terms involving the following variables:

(i) Adoption of the algorithm ($SmartPricing_t$), which is a dummy variable that takes a value of 1 if the focal host/property adopted the algorithm in the given time period;

(ii) Seasonality: For simplicity we classified seasonality in terms of two seasons only: peak and off-peak seasons with peak season being the baseline season. For each city, we classified each month into either peak or off-peak season. We represent this variable by a seasonal dummy, $OffPeak_t$, which takes the value of 1 if the month $t$ was in the off-peak season.

(iii) Number of years lapsed since the launch of the algorithm ($PostYear_t$): If the observation month $t$ in the data lies between Nov 2015 (which is the time period when the algorithm was launched) and Oct 2016, $PostYear_t$ takes a value of 1, and if the observation month $t$ in the data lies between Nov 2016 to Aug 2017, *PostYear* a value of 2. Also $PostYear_t$ =1 includes pre-launch periods.

(iv) *Zip-code adoption rate*$_t$: This is defined the fraction of listings in the same zip-code as that of the focal property, which have adopted the algorithm in the observation month $t$. We operationalize the zip-code adoption rate from our sample.

[16] Please see section 7 of the Web Appendix for details on how we operationalized the peak and off-peak seasons.

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 regression are reported in column 2 of Table S10. The parameters of interest related to the second source are the coefficients of all the interaction terms that involve *zip-code adoption rate* and *SmartPricing*. Observe that the coefficients of these interaction terms are insignificant. This suggests that there may not be significant dynamics in the treatment effects that stem from the second source (the competitive effect). And the parameters of interest related to the first source are the coefficients of all the interaction terms that involve (*PostYear=2)* and *SmartPricing*. Observe that the coefficients of these interaction terms are insignificant. This indicates that there may not be significant dynamics in the treatment effects that stem from the first source (changes in the algorithm over time).

The above result has to be taken with caution for two reasons. First, the average adoption rate of the algorithm across all zip codes by Aug 2017 (which is the last month in our data) is 13.8%, which is small. It could be that the dynamic treatment effects related to competition only kick in at higher adoption rates we do not observe in our data. Second, the variable, *Zip-code Adoption Rate*, may not accurately capture the true adoption rate at the zip code level. This is because we have operationalized the zip-code adoption rate from our sample and not the entire population of listings in each city.

**Table S13 Dynamics Effects of Smart Pricing on Property Daily Revenue: Interacting with Year-Dummies, Off-Peak Season, and Adoption Rate**

| | **ESTIMATES** | |
|---|---|---|
| **VARIABLES** | (1)<br>Interacting<br>Yearly, Off-peak | (2)<br>Interacting Yearly,<br>Off-peak, Zip-code<br>Adoption Rate |
| ***Smart Pricing***<br>(non off-peak season and 1st year as default) | 8.182*** | 10.51*** |
| | (1.738) | (2.623) |
| ***Smart Pricing X Offpeak*** | -2.171 | -6.414 |
| | (3.954) | (4.631) |
| ***Smart Pricing X postYear =2*** | -1.463 | -4.064 |
| | (2.405) | (4.978) |
| ***Smart Pricing X Offpeak X postYear=2*** | -5.996 | -2.055 |
| | (4.794) | (6.147) |
| ***Zip-adoption Rate*** | | -11.44 |
| | | (10.56) |
| ***Smart Pricing X Zip-adoption Rate*** | | -21.91 |
| | | (12.88) |
| ***postYear X Zip-adoption Rate*** | | -9.682 |

| | | |
|---|---|---|
| | | (18.17) |
| ***Smart Pricing X postYear X Zip-adoption Rate*** | | 22.80 |
| | | (27.91) |
| ***Offpeak X postYear X Zip-adoption Rate*** | | 23.24 |
| | | (27.55) |
| ***Smart Pricing X Offpeak X Zip-adoption Rate*** | | 74.51 |
| | | (39.74) |
| ***Smart Pricing X Offpeak X postYear X Zip-adoption Rate*** | | -64.45 |
| | | (50.30) |
| ***Offpeak X Zip-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 |
| Robust Standard errors clustered at individual property level are presented in parentheses p<0.05 ** p<0.01 *** p<0.001 | | |

**6.4 A Comparisons of Alternative Methods: Raw DiD, DiD with Propensity Score Matching (PSM), and DiD with Synthetic Control Method (SCM)**

We present alternative approaches in assessing the estimated impact of using smart pricing on the daily revenue. We first show that a DiD analysis on the raw sample (without using weighting or matching). We then present results from estimating DiD on a PSM-matched sample. Lastly, we employ a Synthetic Control Method (SCM) for estimating the impact of using smart pricing algorithm.

***Performing DiD on Raw Sample (i.e., without Matching/Weighting)***

We report the estimates of the DiD regression on the raw sample in Table S14 below. Since the parallel-trends assumption is rejected on the raw sample (see section 6.1), it follows that doing DiD on the raw sample can yield biased estimates of the treatment effects. This can be seen by comparing the average treatment effect and the differential effect across the DiD on the raw sample and DiD on the IPTW weighted sample. The average treatment effect is 6.396 (7.250) for the IPTW (raw) sample, and the differential effect is 8.700 (9.561) for the IPTW (raw) sample. This shows that performing DiD on the raw sample would lead to an overestimation of both effects.

**Table S14 A pure-DiD Analysis on Raw sample (Without Matching or Weighting Methods)**

| VARIABLES | Main Effect | | Interacting 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 Others* | | | 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 | |
| Cluster-robust standard errors at individual property level in parentheses<br>* p<0.05   ** p<0.01   *** p<0.001 | | | | |

***Performing DiD on PSM-matched Sample***

We report the results of the DiD regression on the matched sample based on PSM in Table S15 below. 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. Note that the average treatment effect of using smart pricing is positive and significant. However, the differential effect of adoption was non-significant (*b*=4.895, p<0.11) for PSM, which is different from the result we obtained using IPTW in which the differential effect was significant.

**Table S15 Impact of Pricing Algorithm on Average Daily Revenues: PSM Analyses**

| VARIABLES | (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×Others* | | | 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 | |
| Cluster-robust standard errors at individual property level in parentheses<br>* p<0.05 ** p<0.01 *** p<0.001 | | | | |

There are two possible reasons for why the differential effect is significant with IPTW and non-significant with PSM. The first possible reason is that PSM and IPTW yield different estimates of the treatment effects – while PSM yields ATT, IPTW yields ATE. The second possible reason is that of low statistical power when using PSM. While constructing the matched sample based on PSM, we had to get rid of 49% of the observations. Thus in our matched sample based on PSM, while we had a total of 5,469 properties with 1,631 adopters, the number of black hosts who adopted the algorithm were only 97. 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.

In order to ascertain which of the two reasons is correct, we ran another DiD regression on a matched sample based on synthetic control method. Note that similar to IPTW, SCM uses a full sample. However similar to PSM, SCM yields the ATT. This implies that the results of DiD on SCM will inform us of which of the two aforementioned reasons holds weight. If SCM were to also yield a non-significant estimate of the differential effect, it would suggest that the reason for the difference in results across IPTW and PSM is not because PSM has low statistical power. Instead it would be because IPTW and PSM yield different types of estimates of the treatment effects. On the other hand, if unlike PSM, SCM were to yield a significant estimate of the differential effect, it would suggest that the reason for the difference in results across IPTW and

PSM is because PSM has low statistical power. This brings us to the DiD regression on the SCM sample.

***Performing SCM Analysis***

The synthetic control strategy (Abadie et al. 2010, Abadie et al. 2015) identifies treatment effect by constructing a synthetic control group of units (i.e., counterfactuals) that mimics the treated units. It is a data-driven approach that, through finding 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-intervention periods. Since SCM constructs a synthetic set that is similar to the treated group, just as PSM, it also yields an ATT. To implement SCM, we used the Generalized Synthetic Control (Xu 2017) method since we have multiple treatment units. We used the R package 'gsynth' developed by Xu and Liu[17] to implement the analysis. The estimation results of SCM are reported in Table S16. Observe that 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. This suggests that the reason why PSM did not yield a significant estimate of the differential effect is because PSM has lower statistical power.

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

| **VARIABLES** | **(1) Main Model** | | **(2) Interacting with Ethnicity** | |
|---|---|---|---|---|
| | **ESTIMATES** | **S.E.** | **ESTIMATES** | **S.E.** |
| ***Smart Pricing*** | 5.471128*** | 1.285069794 | 4.584356*** | 1.375587 |
| ***SmartPricing×Black*** | | | 7.57656* | 3.074063 |
| *SmartPricing×Others* | | | 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 | |
| Cluster-robust Standard errors<br>* p<0.05 ** p<0.01 *** p<0.001 | | | | |

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

## Section 7: Investigating Seasonal Price Correction as an Alternative Mechanism to Explain the Main and Differential Effects of Adoption

In this section, we will investigate whether and to what extent can an alternative mechanism related to seasonal price correction can explain the average treatment effect of adoption (i.e., on an average, adoption of the algorithm leading to increase in revenues of hosts) as well as the differential effect of adoption (i.e., adoption of the algorithm benefiting black hosts more than white hosts). We will explore this alternative mechanism for the average treatment effect in section 7.1 and for the differential effects in section 7.2.

### 7.1 Whether and to What Extent Does Seasonal Price Correction Explain the Average Treatment Effect?

Starting with understanding the mechanism for the average treatment effects, recall that in section 3 of the main paper we regressed both prices and occupancy on adoption of the algorithm. We found that on average, adoption of the algorithm leads to decrease in prices and increase in occupancy rate of hosts, where the increase in occupancy offsets the decrease in prices, which thereby increases revenue. However, we did not investigate whether the downward price correction by the algorithm was the same throughout the year or whether it was more during certain seasons such as the off peak season. One possible reason why the algorithm's downward price correction might be larger during the off-peak season is because the hosts by themselves may not decrease the prices enough when they move into off peak seasons. 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.

To explore these reasons, we first regress property price (average nightly rate) on whether or not the host adopted the smart pricing algorithm in month *t* and its interaction with whether or not month *t* belongs to an off peak season.

$$NightlyRate_{it} = \beta \cdot SmartPricing_{it} + \gamma \cdot SmartPricing_{it} \times OffPeakSeason_{it} + \lambda \cdot Controls_{it} + Seasonality_{it} + Property_i + \varepsilon_{it} \quad \text{(S21)}$$

In this regression, $SmartPricing_{it}$ is an indicator variable that takes the value of 1 if property *i* used the pricing algorithm in month *t* and 0 otherwise. $OffPeakSeason_{it}$ is an indicator variable that takes a value of 1 if month *t* is in an off peak season and 0 if it is not. The parameter β captures the impact of adoption of the algorithm on prices during peak seasons, and the parameter γ captures the differential effect of the algorithm on prices during peak vs. non-peak seasons. For each city, we classified a given month into the peak season or the off-peak season category as follows. For each city, we first calculated the average occupancy rate in each month in our data for the full one year prior to the introduction of the algorithm (i.e., from Nov 2014 to Oct 2015)[18]. Following that,

[18] Please note that we do have the occupancy data from AirDNA 18 months prior to the introduction of the algorithm. However we only used the AirDNA data from July 2015 to Aug 2017 in our analysis. The reason for that is because AirDNA only started collecting information on other dynamic variables (such as number of review, number of photos, property time-varying characteristics etc.) that we use in our analysis from July 2015 onwards.

we categorized the bottom three months with lowest occupancy for each city as the off peak season and the rest of the months as the peak season.

The results of this regression are given in Column 1 of Table S17. The estimate of the main effect β is -6.277 (with $p<0.05$) and the interaction effect γ is -11.01 (with $p<0.05$). These estimates show that there are two mechanisms through which adoption of the algorithm leads to an increase in hosts' revenues: (a) general price correction, whereby the algorithm brings prices down by $6.227 across all seasons during the year, and (b) seasonal price correction, whereby the algorithm brings prices further down by $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.

At this point, it is important to mention that the aforementioned results are robust to the following:

(i) They are robust to the number of months that we include in the peak season and in the off-peak season. In the present analysis, we have included 3 months in the off-peak season and 9 months in the peak season. We have also done the analysis with including 6 months in the peak season and 6 months in the off peak, and also with 9 months in the off-peak season and 3 months in the peak season. The basic nature of results remains the same.

(ii) They are robust to the number of seasons that we consider in our analysis. In the present analysis, we have considered only two seasons – peak and off-peak. We have also done the analysis with 4 seasons, and the basic nature of the results does not change.

(iii) They are robust to the potential limitation of the AirDNA data. Note that the limitation of the AirDNA data is to do with the fact that AirDNA does not observe the true occupancy and it instead predicts it using its algorithm. However in the above regression, we are not working with occupancy. Instead, we are working with price as the dependent variable. As discussed earlier (section 2.5 of Web Appendix), AirDNA has accurate information on each property's listing price. This information is publicly available on each host's website and AirDNA got this information by scraping this information from each host's webpage.

We next explore whether these price corrections lead to increase in occupancy and revenues. We performed two more regressions, which are the same as the aforementioned regression, except that in the first one, the dependent variable is the occupancy rate, and in the second one, the dependent variable is average daily revenue per month. The results of these two regressions are reported in columns 2 and 3 of Table S17 respectively. Starting with column 2, we see that adoption of the algorithm increased occupancy in peak as well as off-peak periods by 0.0710. Thus both price corrections led to increase in occupancy. Moreover note that the increase in occupancy as a result of adoption in the peak seasons and off peak seasons are not statistically different from each other. 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, we see that the increase in revenue is positive during both peak season (=$7.767) and for the off peak season (=$7.767-$3.352=$4.115) as a result of adoption, which shows that both mechanisms lead to an increase in revenue.

In conclusion, the analysis shows that the algorithm increases revenues through both the following mechanisms: (a) general price correction, whereby the algorithm brings prices down by

[19] A possible reason for that is because there are more active Airbnb listings during peak seasons than during off peak seasons (Farronato and Fradkin 2018). This leads to greater competition amongst Airbnb listings during peak seasons, which makes the demand of an individual unit more responsive to prices.

$6.227 across all seasons during the year, and (b) seasonal price correction, whereby the algorithm brings prices further down by $11.01 during the off peak season.

**Table S17 Effects of Using Smart Pricing on the Property Price, Occupancy, and Revenue in the Off-Peak Season**

| | **ESTIMATES** | | |
|---|---|---|---|
| **VARIABLES** | **(1)<br>Nightly Rate** | **(2)<br>Occupancy Rate** | **(3)<br>Daily Revenue** |
| ***Smart Pricing***<br>(non off-peak season as default) | -6.277* | 0.0710*** | 7.467*** |
| | (3.094) | (0.00558) | (1.492) |
| ***Smart Pricing X Off Peak*** | -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*** |
| | (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 |
| Cluster-robust standard errors at individual property level in parentheses<br>* p<0.05 ** p<0.01 *** p<0.001 | | | |

**7.2 Whether and to What Extent Does Seasonal Price Correction Explain the Differential Effect?**

Moving on to understanding the mechanism for the differential effect, recall that in section 3 of the paper we regressed both prices and occupancy on the two-way interaction between adoption of the algorithm and ethnicity of the host. We found that while adoption of the algorithm led to a similar magnitude of downward price correction across black and white hosts, it led to a much greater increase in occupancy for black hosts as opposed to white hosts. Based on this result we concluded that the reason why black hosts benefitted more than white hosts 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.

However, there can be an alternative explanation that does not require the demand curves for black and white hosts to be different. The alternative explanation is that even though the downward price correction (averaged over the entire year) is the same across black and white hosts, it could be larger for black hosts as compared during the seasons when the Airbnb demand is more responsive to prices, 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. In this section, we will explore this alternative mechanism.

To explore the different mechanisms, we extend the regressions in equation (S21) by adding a three-way interaction involving host's ethnicity. We start with the regression (equation S22) in which price is the DV, where we categorize ethnicity by *White, Black* or *Others*, and set *White* as the reference group:

$$\begin{aligned} NightlyRate_{it} &= \beta \cdot SmartPricing_{it} + \beta_1 \cdot SmartPricing_{it} \times Ethnicity_i + \beta_2 \\ &\cdot SmartPricing_{it} \times OffPeakSeason_{it} \times Ethnicity_i + \gamma \\ &\cdot SmartPricing_{it} \times OffPeakSeason_{it} + \gamma_1 \cdot OffPeakSeason_{it} \\ &\times Ethnicity_i + \lambda \cdot Controls_{it} + Seasonality_{it} + Property_i + \varepsilon_{it} \end{aligned} \tag{S22}$$

The results are reported in column 1 of Table S18. The estimates of the coefficients of *SmartPricing* (=-7.411) and *SmartPricing*×*OffPeakSeason* (=-12.09) are statistically significant at $p<0.05$, which implies that adoption of the algorithm led to a significant downward general as well as seasonal price correction for white hosts.

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, *Off Peak Season*×*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 the interaction term, *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*×*OffPeakSeason*, is also not statistically significant, which implies that adoption of the algorithm led to a similar magnitude of downward price correction between black and white hosts during off-peak season.

Thus we see that across all seasons of the year, there is no significant difference in the extent of downward price correction by the algorithm between black and white hosts. This shows that differences in seasonal price correction between black and white hosts does not explain the differential effect of the algorithm. At this point, it is important to mention that the aforementioned results are robust to the following:

(i) They are robust to the number of months that we include in the peak season and in the off-peak season. In the present analysis, we have included 3 months in the off-peak season and 9 months in the peak season. We have also done the analysis with including 6 months in the peak season and 6 months in the off peak, and also with 9 months in the off-peak season and 3 months in the peak season. The basic nature of the results does not change.

(ii) They are robust to the number of seasons that we consider in our analysis. In the present analysis, we have considered only two seasons – peak and off-peak. We have also done the analysis with 4 seasons, and the basic nature of the results does not change.

(iii) They are robust to the potential limitation of the AirDNA data. Note that the limitation of the AirDNA data is to do with the fact that AirDNA does not observe the true occupancy and it instead predicts it using its algorithm. However in the above regression, we are not working with occupancy. Instead, we are working with price as the dependent variable. As discussed earlier, AirDNA has accurate information on each property's listing price. This information is publicly available on each host's website and AirDNA got this information by scraping this information from each host's webpage.

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. To do so, we run the same regression as in equation (S22), except that the DV is now occupancy rate. The results of this regression are reported in column 2 of Table S12. The estimate of the coefficient of *SmartPricing* is 0.0639 with p<0.01. The estimates of the coefficients all the interaction terms involving *Ethnicity* are statistically non-significant, except that of the interaction term, *SmartPricing*×*Black*. The estimate of the coefficient of this term is 0.0763 (with p<0.01). Thus we see that adoption of the algorithm increased the occupancy of white hosts by 0.0639 and that of black hosts by 0.1402 (=0.0639+0.0763). This shows that the reason why adoption of the algorithm benefitted black hosts more than white hosts is because the demand of black hosts is more responsive to price changes as compared to that of white hosts.

**Table S18 Effects of Smart Pricing on Property Price and Occupancy: Ethnic Groups of Hosts in Off-peak Season**

| | **ESTIMATES** | |
|---|---|---|
| **VARIABLES** | **(1) Nightly Rate** | **(2) Occupancy Rate** |
| ***Smart Pricing*** (non off-peak season as default) | -7.411* | 0.0639*** |
| | (3.659) | (0.00608) |
| *Smart Pricing X Off Peak* | -12.09* | -0.0140 |
| | (5.744) | (0.00819) |
| ***Smart Pricing X Black*** (white ethnicity as reference) | 5.003 | 0.0763*** |
| | (5.388) | (0.0202) |
| *Smart Pricing X Other* | 6.875 | 0.0205 |
| | (4.985) | (0.0156) |
| ***Off Peak X Black*** | -0.586 | -0.0181 |
| | (1.028) | (0.00988) |
| *Off Peak X Other* | 3.073 | 0.00390 |
| | (2.690) | (0.00935) |
| ***Smart Pricing X Off Peak X Black*** | 15.00 | -0.0195 |
| | (8.112) | (0.0289) |

| | | |
|---|---|---|
| *Smart Pricing X Off Peak 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 |
| * p<0.05 ** p<0.01 *** p<0.001<br>Cluster-robust standard errors at individual property level in parentheses | | |

## Section 8. Exploring Policy Implications

We present analyses for two policy recommendations for Airbnb that would help reduce the revenue disparity between the 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 are correlated with race into the pricing algorithm

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.

In order to know which non-race variables are correlated with 'Black', we performed a logistic regression in which we regressed *Black* on the entire set of covariates that we had used in our propensity score estimation. We report the results in Table S19 below. Some of the key covariates that are significantly correlated with *Black* are: (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 in a month set by the host, (vi) number of minutes to drive to downtown, (vii) transit score (which captures how well the area near the property is served by public transportation), and (viii) whether or not the property had a heating and a refrigerator. This would suggest that these adding these variables in the algorithm can reduce the racial gap.

However, there are two caveats in place here. *First*, the McFadden's pseudo $R^2$ for this logit regression is 0.1. Although it not very low[20], is not large either. This suggests that while including these covariates in the algorithm will reduce the revenue gap, however the reduction will be much smaller than if Airbnb were to include race in its algorithm. *Second,* including these socioeconomic variables in the algorithm can also run into legal problems because it will result in disparate impact across different races (Barocas and Selbst. 2016). Note that both disparate treatment by the algorithm (which results from including race in the algorithm) as well as disparate impact by the algorithm (which results from including socioeconomic characteristics that are correlated with race in the algorithm) are illegal.

**Table S19 Regressing Race Variable (Black) on Covariates**

| VARIABLES | 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) |
| # Minimum Stay | -0.00683 | (0.0146) |
| Security Deposit | -0.000165 | (0.000134) |

[20] The reason why the pseud R2 of 0.1 if not very low is because black hosts only constitute 10% of the hosts in the population. It is difficult to get high values of pseudo $R^2$ with such extreme proportions of ethnicities in the population,

| | | |
|---|---|---|
| # Blocked Days in a 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 | |
| Note: the D.V. is a binary variable *Black*, the model is a logits regression. The estimation was done on sample excluding properties with *Others* ethnicity. The same set of variables used in the propensity score model is used.<br>Standard errors in parentheses. * p<0.05 ** p<0.01 *** p<0.001 | | |

### 8.2. Targeting/encouraging black hosts to adopt the pricing algorithm

To identify which socioeconomic segment(s) of black hosts should Airbnb target in order to encourage them to adopt, we segmented the hosts into four quartiles based on their socioeconomic status. We use education as a measure of the socioeconomic status, which we operationalize as the percentage of adults within the same zip-code with the same ethnicity who have a bachelor's degree. We collected the information on the percentage of adults with a graduate and bachelor's degree from US Census data (ACS—American Community Survey).

We assigned each property into one of the following four quartiles, where the quartile categorization were was done separately for each city: quartile q=1 as the bottom quartile (lowest end in the distribution of education), quartile q=2 as the second from the b7ottom bucket, quartile q=3 as the second from the top bucket, and quartile q=4 was the top quartile (highest end in the distribution of education). This yields a four quartile categorization, *q*=1..4, for education. Next we did the following analysis:

(a) We performed a separate IPTW+DiD regression for each quartile of education. In the DiD regression for each quartile, we regressed revenue over a two-way interaction between adoption and race.[21]

(b) We performed a logit regression, where we regressed adoption of the algorithm over a two-way interaction between the different quartiles of education and ethnicity, where controlling for all other variables that we had used in the adoption model.

The results of the DiD regression in (a) are reported in Table S19 below. The main effect of adoption (i.e., the coefficient of *SmartPricing*) is positive and significant for quartiles q=1, 2 and 3, but not significant for the topmost quartile q=4. The differential effect of adoption (i.e., the coefficient of *SmartPricing*×*Black*) is positive and significant for q= 3 only. The magnitudes of these effects imply that amongst the black hosts, the treatment effect of adoption is the highest for quartile q=3 of education, followed by quartiles q=1 and 2, and is non-significant for q=4. A possible reason why the treatment effect is non-significant for q=4 is that these hosts are in the upper most quartile of education and would as such be proficient in pricing their properties – thus adopting the algorithm does not lead to any significant increase in revenues for them.

We next discuss the logit regression in (b). The results are reported in Table S20 below. The main effect of Black is negative and significant. However most of the interactions of Black with the quartiles of either of the two socioeconomic variables are not significant – the only significant one is the interaction of Black with the upper most quartile q=4. Based on the magnitudes of these effects, we see that as far as the adoption rate amongst black hosts is concerned, it is highest for quartile q=1 of education, followed by q=2, then followed by q=3, and finally followed by q=4.

There are two implications that follow from the above discussion. *First*, even though black hosts in q=4 have the lowest rate of adoption, it does not make sense for Airbnb to target them in order to encourage them to adopt the algorithm. This is because black hosts in quartile q=4 do not stand to monetarily gain by adopting the algorithm.[22] *Second*, amongst the other three quartiles of education, the adoption of black hosts is lowest in q=3, but black hosts in this quartile stand to gain the most by adoption. This implies that it is especially important for Airbnb to target black hosts

[21] We also performed a single IPTW+DiD regression with all four quartiles together, where we regressed revenue over a three-way interaction of the different quartiles of the given socioeconomic variable, race and adoption. The results were qualitatively similar to the results presented here.

[22] A possible reason why hosts in this quartile may be adopting the algorithm is because of the convenience that they do not have to decide and manually set prices every day.

in q=3 in order to encourage them to adopt. The other two quartiles q=1 and 2 of black hosts are also important to target, although to lesser extent as compared to q=3.

**Table S19 Impact of Pricing Algorithm on Average Daily Revenues: A separate IPTW and DiD regression for each quartile of % of Bachelor's Degree**

| VARIABLES | Quartile: q=1 (bottom) | Quartile: q=2 (2nd bottom) | Quartile: q=3 (2nd top) | Quartile: q=4 (top) |
|---|---|---|---|---|
| ***Smart Pricing*** | 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×Others* | 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 |

Note: These are the IPTW and DiD regression results for each quartile of the given socioeconomic characteristic. The categorization into the four quartiles was performed within each city separately. The number of observations vary because 1) number of units used in each regression, and 2) observations where a property that had full month blocked in a month are automatically dropped from regressions (as the revenue=price*occupancy would be undefinable).
Cluster-robust standard errors at individual property level in parentheses
* p<0.05 ** p<0.01 *** p<0.001

**Table S20 Impact of Race and its Interaction with the different Quartiles of Education on Adoption of the Algorithm**

| | **q=Education (% bachelor) as quartiles Estimate (Std. Error)** | |
|---|---|---|
| black | -0.535*** | (0.148) |
| others | -0.227 | (0.160) |
| q=2 | -0.162* | (0.0822) |
| q=3 | -0.257** | (0.0831) |
| q=4 | -0.128 | (0.0875) |
| black # q=2 | -0.106 | (0.244) |
| black # q=3 | 0.0739 | (0.272) |
| black # q=4 | -1.100** | (0.377) |
| others # q=2 | -0.120 | (0.235) |
| others # q=3 | -0.127 | (0.244) |
| others # q=4 | -0.488 | (0.255) |
| Bedrooms | -0.0819 | (0.0460) |
| Apartment | -0.167* | (0.0708) |
| EntirePlace | -0.356*** | (0.0678) |
| Title_Length | 0.0129** | (0.00410) |
| NumberofPhotos | 0.00716** | (0.00247) |
| NumberofReviews | 0.00000811 | (0.000960) |
| NightlyRate | -0.000101 | (0.000386) |
| MinimumStay | -0.00257 | (0.0116) |
| SecurityDeposit | 0.0000367 | (0.0000919) |
| BlockedDays | -0.0268*** | (0.00295) |
| ReservationDays | 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) |
| smoke_detector | 0.332** | (0.102) |
| Shampoo | -0.0459 | (0.0853) |
| Breakfast | 0.0273 | (0.110) |
| Ac | 0.137 | (0.571) |
| num_faces_mean | 0.0190 | (0.0361) |
| WalkScore | -0.00278 | (0.00165) |
| TransitScore | 0.00751*** | (0.00227) |
| DriveTime | -0.00165 | (0.00300) |
| Population Density (Per Sq. Mile) | 0.00000025 | (0.00000125) |
| age_mean | -0.00706* | (0.00278) |
| listing_month | -0.00469* | (0.00207) |
| num_listings | -0.0392*** | (0.00769) |
| 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).