Article

Web Scraping for Hospitality Research: Overview, Opportunities, and Implications

Saram Han[1,2] and **Christopher K. Anderson[1]**

Abstract

As consumers increasingly research and purchase hospitality and travel services online, new research opportunities have become available to hospitality academics. There is a growing interest in understanding the online travel marketplace among hospitality researchers. Although many researchers have attempted to better understand the online travel market through the use of analytical models, experiments, or survey collection, these studies often fail to capture the full complexity of the market. Academics often rely upon survey data or experiments owing to their ease of collection or potentially to the difficulty in assembling online data. In this study, we hope to equip hospitality researchers with the tools and methods to augment their traditional data sources with the readily available data that consumers use to make their travel choices. In this article, we provide a guideline (and Python code) for how to best collect/scrape publicly available online hotel data. We focus on the collection of online data across numerous platforms, including online travel agents, review sites, and hotel brand sites. We outline some exciting possibilities regarding how these data sources might be utilized, as well as discuss some of the caveats that have to be considered when analyzing online data.

Introduction

The internet presents many interesting opportunities for understanding consumer choice. Any information displayed on any travel website represents a potential data source for hospitality researchers. For instance, a hospitality researcher might collect a list of reviews from TripAdvisor and perform text mining, or retrieve satisfaction ratings to perform some statistical analysis, or determine the relationship between price and page ranking at an online travel agent (OTA). Despite the appeal of these opportunities, however, the fact remains that manually collecting a large amount of online data is inefficient and practically impossible. A typical hospitality research study may require data from multiple hotels across different markets over a prolonged period of time; collecting this data manually can be incredibly time consuming and tedious. Therefore, hospitality researchers need to automate the process of gathering and storing the information presented on travel websites. This is where web scraping comes into play. Web scraping is the process of creating a computer program to download, parse, and organize data from the web in an automated manner (vanden Broucke & Baesens, 2018).

As Table 1 illustrates, there are several alternative approaches that hospitality researchers may consider when trying to automate the collection of online data. One promising option is using an Application Programming Interface (API) provided by the firm hosting the data. However, APIs are often difficult to access, accessible only for a limited duration, or require upfront fees even when researchers are able to access an API. Furthermore, APIs generally will not expose all the required variables.

Recently, scholars have utilized alternative approaches to collect consumer behavior data that mimics online behaviors. Previous studies directly captured data in a simulation lab or survey setting where research participants acted as if they were actually completing online activities (Choi et al., 2017; Min et al., 2015; Shin et al., 2019). However, a common criticism of these studies is that there is a mismatch between real customer behavior and the behavior of the subjects in these experiments, which are often conducted by undergraduate students or participants in Amazon's Mechanical Turk (Schahn & Holzer, 1990). Moreover, although it is possible to conduct large-scale

[1]Cornell University, Ithaca, NY, USA
[2]Seoul National University of Science and Technology, Republic of Korea

Corresponding Author:
Saram Han, Seoul National University of Science and Technology, 232 Gongneung-ro, Nowon-gu, Seoul 01811, Republic of Korea.
Email: saramhan@seoultech.ac.kr

Table 1.
Comparison of Common Data Collection Methods in Online Market Research.

	Scraped Data	Commercial Web Scraping Service	API	Survey
Cost	Low	Medium	Low/Medium	High
Sample frame	Website users	Website users	Website users	Flexible
Customizability of variables	Medium	Low	Low	High
Ease of frequent collection	Easy	Moderate	Easy	Hard
Data type	Behavioral	Behavioral	Behavioral	Attitudinal
Limitations	Time and programming skills	Data may not be suitable to the researcher's need in terms of variables or content	Limited availability	Time and programming skills

Note. API = Application Programming Interface.

survey research, a number of barriers make that process difficult. Meanwhile, the scale of web scraped data usually is large in nature. Compared with conducting a survey, collecting publicly available online data is inexpensive—even considering the required fee for the commercial service. The monetary cost and effort required to conduct a survey increases linearly with each additional sample, whereas web scraping requires only a one-time expense. Hotel prices change daily, and online reviews are posted even more frequently. In such a dynamic market, a single sampling may not be sufficient. Conducting multiple surveys sequentially is expensive and requires a lot of effort, whereas collecting publicly available online data enables researchers to extract as much data as they need in real time.

Another data collection method that researchers frequently chose is to outsource the scraping task to third-party commercial firms (Wu et al., 2015). Although this is perhaps the simplest option, there are potential pitfalls that may arise due to the lack of discretion in the data collection process. For instance, researchers often realize too late that they forgot to ask the firm to scrape for certain variables. As a result, they are forced to either spend additional time, effort, and money or complete their research without having access to all the desired variables or behaviors.

However, the existing literature points out several drawbacks to web scraped data. One common critique of online data is that study results derived from this data are rarely generalizable to the behavior of the entire target population; although these data are useful to researchers who are interested only in the behavior of online customers, if the researcher's target population includes both online and offline customers, any results derived from online data will be highly vulnerable to selection bias. This is because web scraped data may overrepresent the unique behaviors of online customers. Meanwhile, researchers who opt to utilize surveys as their research method have the flexibility to design a sample frame capable of representing both online and offline customers. Another drawback of publicly

available data is that researchers have little discretion in measuring aspects from the sample. In contrast, researchers can include anything they want in a survey or questionnaire, and thus have no restriction in the variables that they can collect. Researchers should carefully decide whether web scraping is suitable to their research and fully consider the potential challenges web scraping poses. An ideal approach might be to collect data through both experiments and web scraping to confirm both the internal and external validity (Viglia & Dolnicar, 2020). For example, some researchers use web scraped data to develop their hypotheses and use traditional data collection methods to confirm these hypotheses (Kupor & Tormala, 2018). Although web scraping has become popular recently and presents an important opportunity to better understand the online marketplace, traditional data collection methods remain the most commonly used strategies in hospitality research.

Table 2 illustrates the potential benefits of web scraped data for hospitality research by summarizing papers by data type with a focus on online reviews in travel marketplaces. To gather this data, we surveyed six highly reputable hospitality journals using the advanced search feature that is available on any journal website. Our analysis included all papers written prior to July 2020. We classified papers as having utilized traditional data collections methods (such as surveys or interviews) if they included the term "online review" and at least one of the terms {"interview," "survey," "amazon mechanical turk," "questionnaire"}, but none of the terms {"scrape," "scraping," "crawl," "actual review," "python"}. In contrast, any papers including the term "online review" and at least one of the following terms {"scrape," "scraping," "crawl," "actual review," "python"} were classified as having directly used web data for online review research. The search queries we used are as follows:

- Interview or survey: "online review" AND ("interview" OR "survey" OR "amazon mechanical turk" OR "questionnaire") NOT ("scrape" OR "scraping" OR "crawl" OR "actual review" OR "python").

Table 2.

Data Collection Methods Used for Online Review Research in Hospitality Journals.

	Interview OR Survey	Web Data
Cornell Hospitality Quarterly	16	3
Journal of Travel Research	21	4
Journal of Hospitality and Tourism Research	7	3
International Journal of Hospitality Management	138	21
Annals of Tourism Research	31	5
Tourism Management	82	25

- Web data: "online review" AND ("scrape" OR "scraping" OR "crawl" OR "actual review" OR "python").

As the table indicates, only a handful of papers use data from the internet to tackle research questions that are related to online marketplaces.

The increasing availability of open source web scraping tools has made it a lot easier for the researchers to build their own customized web scrapers. This article aims to dismantle some of the barriers that hospitality researchers encounter when attempting to utilize web scraping in their online studies. However, this article is not a complete guide to web scraping, but rather an introduction to some of the key tools and requirements for scraping key hospitality data sources. Our article provides hospitality researchers with tools and techniques for collecting data from typical, interactive hotel websites. Due to the introductory nature of this article, we focus on applied concepts and functional illustrations. In this article, we assume that readers have fundamental knowledge of programming languages, such as defining variables, loops, and functions, but not necessarily Python. Proficient programmers and researchers who are familiar with web scraping may already be capable of writing their own code from scratch, and thus may find this article to be too applied or specific. The methods we suggest may not represent the most efficient methods available, as this article only considers websites that have dynamic contents—that is, webpages designed to change in response to human interactions (Massimino, 2016): for example, say a researcher wants to scrape not only all the customer reviews on a particular hotel webpage on TripAdvisor but also the profile information from each individual reviewer. TripAdvisor's dynamic website is designed to present detailed profile information only when a user positions their cursor over the reviewer's profile picture. This is where the dynamic-website specialized code that we cover in this article becomes useful. In contrast, if the website contains only static contents—meaning that the website does not change until the user (i.e., client) moves on to another webpage—our dynamic-website specialized code may take longer to function than static-website specialized code. Therefore, for efficient coding, some researchers may want to refer to Massimino (2016) if the website contains only static contents. While the dynamic-website specialized codes can be used to analyze both static and dynamic contents, static-website specialized codes cannot analyze dynamic contents. As the goal of this article is to provide hospitality researchers with a generally useful tool that can be easily applied to a variety of websites, we only consider the dynamic-website specialized code in this article. Popular Python libraries specialized for static website scraping are Requests, BeautifulSoup, lxml, and Scrapy. Those who are interested in detailed instructions from utilizing these libraries and other web scraping methods that can be applied to broader websites should read vanden Broucke and Baesens (2018). We hope our article will make online data collection easier for hospitality researchers and spark greater interest in web scraping techniques. Therefore, our purposes include the following:

- Providing tools applicable to major hotel platforms (i.e., TripAdvisor, Expedia, Marriott.com, and Airbnb),
- Introducing the process in the simplest terms possible, and
- Discussing both the benefits and limitations of analyzing web scraped data.

This article is a step-by-step guide to web scraping, with a focus on websites that are particularly useful to hospitality researchers. Other papers tend to either focus too narrowly on specific websites or discuss web scraping at an abstract level. Considering that different travel websites provide different insights, hospitality researchers rarely utilize only one specific website in their research. Narrowly focused articles require hospitality researchers to learn different web scraping methods for each individual website included in their research. However, overly abstract articles often fail to provide all-in-one, generalized information that novices can apply to their own research. To narrow the gap between the needs of hospitality researchers and the currently available resources, our article focuses on two aspects that have not previously been thoroughly examined in a single paper.

First, rather than introducing different tools for each individual website, we introduce more general tools that are applicable to all major travel websites. Second, after reading our article, researchers will have all the knowledge they need to start web scraping, as our article provides the entire scripts used for scraping every major travel website.

The remainder of this article is constructed as follows. First, we introduce the environments required for running a Python-based scraper. Then, we explain the sample web scraping code for collecting online hotel reviews and prices. In addition to providing samples of key code required for scraping through the paper, in the supplemental appendix we provide complete scripts for scraping reviews and prices from major OTAs, review sites and hotel brand sites. As most web scraped data are secondary data, with the data generation process outside the researcher's control, biases have to be handled with caution. Therefore, we also illustrate some possible biases that researchers should be aware of before analyzing scraped data. Finally, we discuss the academic implications of web scraping and the ethical issues that hospitality researchers should consider when collecting online data for their own research.

Web Scraping Fundamentals

In the following section, we outline the fundamentals of web scraping illustrated through the use of Python—a general-purpose programming language. Our article is not an introduction to Python as a whole, but rather focuses on the aspects of Python that are key to web scraping. Python is a very approachable and intuitive programming language and is often the language of choice for many introductory programming courses. There are many online resources that provide an overview of Python. For those who are interested in learning the basics of Python language, we recommend reading Downey (2014). However, our focus in this section is on the unique methods and skills necessary for scraping hospitality data from a variety of sources using Python. As there is a large scraping community that uses Python3, throughout the article we utilize Python3 specifically (https://www.python.org/downloads/).

Required Environment and Python Codes

As our goal is to simplify the process of gathering online information and transforming it into a meaningful data set, we focus only on the essential elements of web scraping in this article. Over the course of this article, we use three different tools to make a scraper that facilitates the collection of data from major hotel websites: Selenium, WebDriver, and XPath. Figure 1 presents the flowchart of the steps of how we utilize the tools specialized for web scraping the dynamic contents.
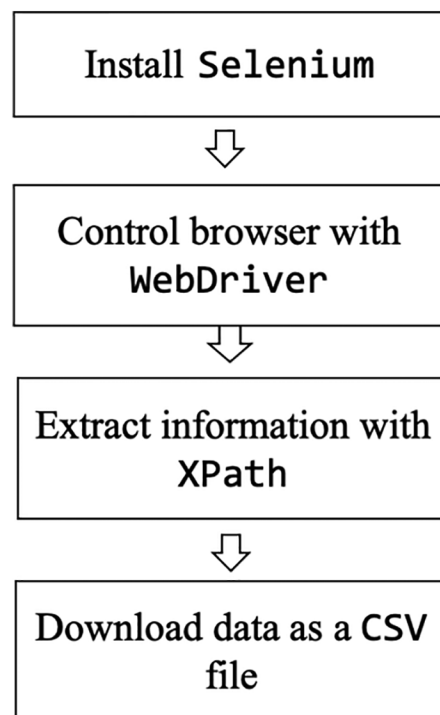
Figure 1.
Flowchart of Web Scraping for Extracting Dynamic Contents.

Selenium. In this section, we introduce the Python library Selenium. Although there are many libraries in Python that facilitate web scraping, Selenium is the most useful when dealing with interactive, JavaScript-heavy pages like those on travel sites such as TripAdvisor, Airbnb, and Expedia. We start by illustrating scraping approaches for extracting review data from TripAdvisor (Listing 1). We then extend this approach to extract review data from other platforms and later extend this approach further to collect other types of data, for example, prices. Let's begin by scraping reviews and basic reviewer profile information from the following TripAdvisor link: https://www.tripadvisor.com/Hotel_Review-g60763-d93344-Reviews-The_Watson_Hotel-New_York_City_New_York.html.
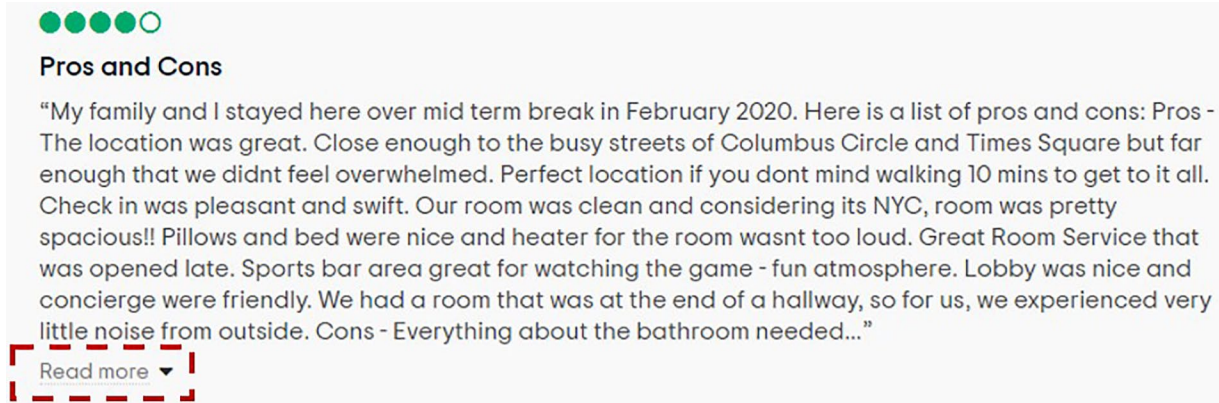
Listing 1.
Define Our Target Website for Scraping.

```
1.   target_page = "https://www.tripadvisor.
     com/Hotel_Review-g60763d93344-
     Reviews-The_Watson_Hotel-New_York_
     City_New_York.html"
```

First, let's name this target page. With other Python libraries (BeautifulSoup, request, Scrapy, etc.), you can

Figure 2.
Example of an Interactive Platform.

scrape everything that is displayed on the screen. While this is good news, it also means that you cannot scrape information that is hidden until you click on it. For instance, contents that are hidden behind the button "Read more," as illustrated in Figure 2, cannot be scraped with other popular Python libraries such as BeautifulSoup, request, or Scrapy. Unlike websites that are only written in Hypertext Markup Language (HTML) or CSS, many interactive travel sites use JavaScript, which allow for interactive functionality on the web page. Contrary to many other programming languages, the core functionality of JavaScript lies in making webpages more interactive and dynamic. Therefore, for sites that make heavy use of JavaScript, we need to write our code in a way that emulates human browsing behavior. That is where the Python library Selenium comes into play. You can easily install it with the following code in the command line: `pip install -U selenium` from the terminal window on your computer. Selenium requires a third-party software called a `WebDriver` which we discuss in the next paragraph.

WebDriver. WebDrivers exist for most modern internet browsers, including Chrome, Firefox, Safari, and Internet Explorer. When using these browsers, a browser window will open up on your screen and perform the actions specified in your code. We can easily download a WebDriver from https://sites.google.com/a/chromium.org/chromedriver/downloads. As WebDrivers exist for most popular browsers, you can choose to download whichever WebDriver best works for you. In the following section, we assume that readers will download the WebDriver for Google's Chrome browser—chromedriver. Make sure you download the file that matches both your operating system (i.e., Windows, Mac, or Linux) and the version of your current browser (i.e., Chrome, Internet Explorer, Edge, or Firefox). The ZIP file you download will contain an executable called chromedriver.exe on Windows, or simply

chromedriver otherwise. While locating the WebDriver in the same directory as your Python scripts is the easiest way to call the driver, it is also possible to explicitly pass the location of where you have the WebDriver as we do with the variable `dirver_loc` in the code Listing 2. Once the WebDriver opens the browser, you can interact with the browser until you close it (i.e., `driver.close()`).

Listing 2.
Python Code for Opening and Closing Your WebDriver.

```
1.  driver_loc = 'C:/Users/Desktop/chrome
    driver.exe'
2.  driver = webdriver.Chrome(driver_loc)
3.  driver.get(target_page)
4.  #Write your scraper here.
5.  driver.close()
```

XPath. Once a page is loaded with WebDriver, you will want to extract the information displayed on the screen and download it on your computer. To extract this information, you first need to view the source code of the website elements you want to scrape. For Windows users, the easiest way to check this information is to right-click on the part within the web page you want to scrape and select "Inspect." You will then see a tree-based view of the HTML codes. Every web page is made up of a bunch of these HTML tags denoting each type of content on the page. By clicking on the arrows, you can see the nested structure of the code. To scrape hotel review websites, we only need to understand the basic structure of HTML language, which consists of tag, attribute, and the value of the attribute. Figure 3 illustrates part of the source code underlying the page displayed after selecting "Inspect."

As an illustration, let us assume that the element of the HTML code written to display the reviewer's name (i.e.,
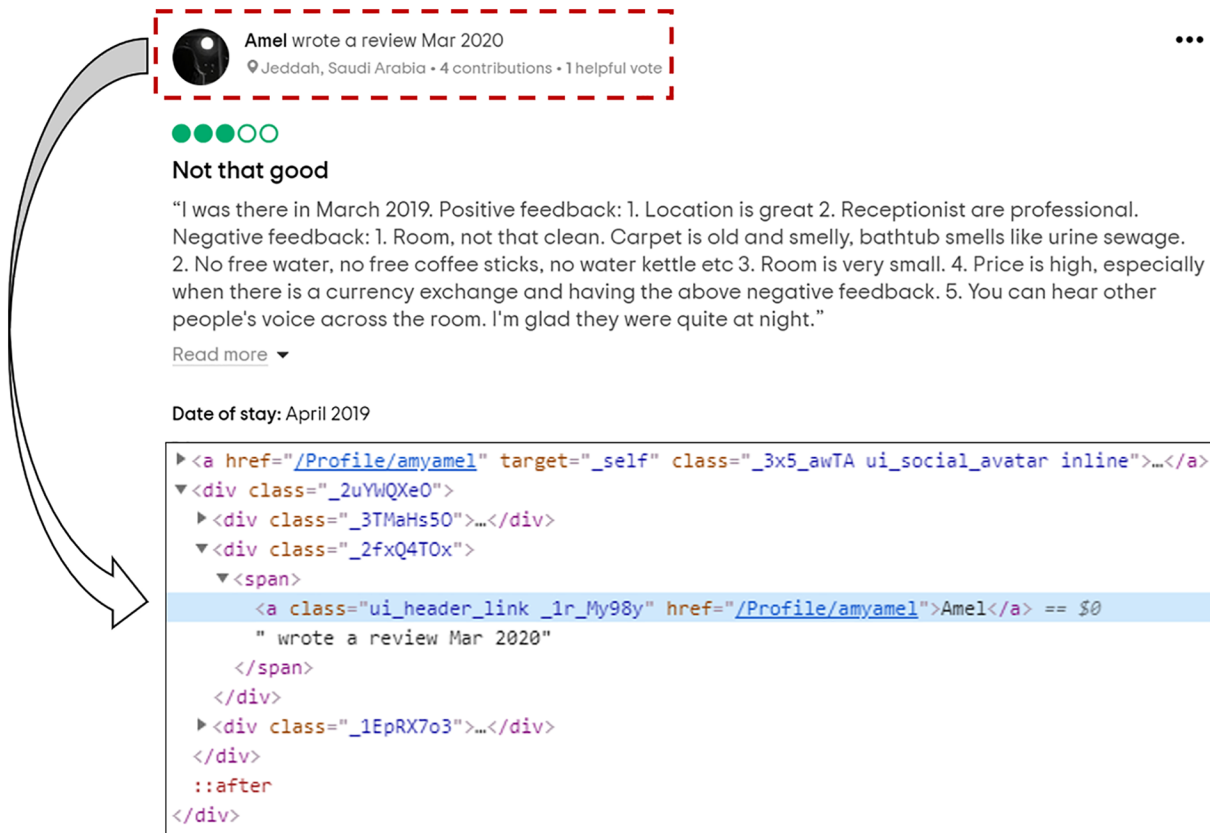
Figure 3.
Displayed Tree-Based HTML Code View after Clicking on Inspect.

Amel) is shown in Listing 3, where the `<a>. . .</a>` is just one of the many existing tags (e.g., `<p>`, `<div>`, `<ul>`) in HTML which encloses the hyperlink and styles the web designer decided to use when displaying the reviewer's name. The value of the attribute "`MemberBlock`" is the name of a style that is assigned to the attribute class. Likewise, `href` is another attribute of this element that redirects users to the reviewer's profile page when they click on the reviewer's username. XPath is a language used to find the location of any element on a web page using this structure of the HTML. Selenium can use the XPath language to select elements. Although there are many Selenium methods, we use only find_element_by_xpath and find_elements_by_xpath throughout this article for ease of application.

Listing 3.
HTML Element for the Reviewer's Name.

```
1.  <a class="MemberBlock" href="/Profile/
    amyamel">Amel</a>
```

To extract the information from the HTML code Listing 3, the XPath has to be written as, `//a[@`

`class='MemberBlock']`, which basically is `//Tag[@ Attribute='Value']` that can be interpreted as "grab the current node (`//`) with the tag name where the class attribute corresponds to `MemberBlock`." However, oftentimes the value of the attribute is very long and contains multiple special characters. This can be difficult because if even one of the characters is missing, a serious error may occur. `//a[contains(@class, 'Member')]` presents an easier way to call the same element, by utilizing the same process but not requiring users to include the entire value of the attribute. This XPath prevents the researcher from provoking an error simply by not indicating the exact value of the attribute. The function `contains()` enables Selenium to easily find the desired element, even if the researcher only provides a fragment of the attribute value. Accordingly, the `find_element_by_xpath` method in the code Listing 4 indicates the element of the HTML code Listing 3.

Listing 4.
Indicating the HTML Element.

```
1.  driver.find_element_by_xpath(".//a
    [contains(@class, 'Member')]")
```

To extract the reviewer's name, we can add .text as shown in the code Listing 5.

Listing 5.
Indicating the Text Content.

```
1.  driver.find_elements_by_xpath(".//a
    [contains(@class, 'Member')]").text
```

Note that the value of the same attribute can always change (e.g., `MemberBlock`) whenever the web designer defines a different name for the value. Therefore, the same web scraper may not work if the web designer updates the HTML code. If an error occurs after running the codes that we provided in the supplemental appendix, we recommend our readers to review the HTML code of the target website and check whether the value of the attribute has changed.

If the researcher is interested in extracting the user's profile link, the following `.get_attribute("href")` method is useful for finding the value of the `href` attribute (Listing 6).

Listing 6.
Indicating the Value of the `"href"` Attribute.

```
1.  driver.find_element_by_xpath(".//a
    [contains(@class, 'Member')]").text.
    get_attribute("href")
```

Applications for Major Hotel Platforms

In this section, we apply these scripts to scrape four major hotel review websites and discuss how to best deal with platform-specific features when building a scraper. We also discuss how hospitality researchers can take advantage of the unique opportunities presented by each individual platform.

Reviews on TripAdvisor. TripAdvisor is the largest player in the travel review platforms arena (Wang & Chaudhry, 2018). Therefore, this platform has been used in many marketing and hospitality studies (Chevalier et al., 2018; Wang & Chaudhry, 2018). TripAdvisor has several unique features that make it attractive to researchers. For instance, each individual's previous platform activities are displayed on their profile. This feature allows researchers to take reviewer level heterogeneity into account as they analyze TripAdvisor data (Gao et al., 2017). Therefore, we recommend scraping each reviewer's profile URL to collect individual-level data that may be useful in the future.

Another interesting feature that recent studies have started to focus on is the existence of *retailer-prompted reviews* (Askalidis et al., 2017; Han & Anderson, 2020;

Mayzlin et al., 2014). Unlike regular self-motivated reviews, retailer-prompted reviews are those that are posted in response to hotels' email invitations to post reviews online. This feature allows researchers to investigate how rating behavior differs depending on the nature of the review posting process.

Although most scraping tasks can be completed using the aforementioned functions and XPath, there is one problem that requires additional attention. As mentioned in Figure 2, many TripAdvisor reviews are long, meaning that users must click on the "Read more" button to see the full review. Researchers must be able to automatically click this button to scrape entire reviews. For this purpose, Selenium offers a selection of "actions" that can be performed by the browser, such as clicking elements. Fortunately, in TripAdvisor, once you expand one review, the rest of the reviews within the same page expand as well. If for some reason all of the desired reviews do not automatically expand with a single click, a for-loop can be added to expand each review individually (see Listing 8). Once we identify the element where the "Read more" button locates, we can use the `execute_script` method to expand the review. We recommend using try-except to avoid the scraper stopping in situations where there is no expandable review on the page.

Listing 7.
Python Script to Expand the Review Element.

```
1.  more  =  driver.find_elements_by_xpath
    (".//div[contains(@data-test-target,
    'expand')]")
2.  try:
3.      driver.execute_script("arguments[0].
        click();", more[0])
4.  except:
5.      pass
```

Listing 8.
Python Script to Expand Every Review Element Within a for-loop.

```
1.  for container in containers:
2.      more = container.find_elements_by_
        xpath(".//div[contains(@class,
        'expand')]")
3.      try:
4.          driver.execute_script("arguments
            [0].click();", more[0])
5.      except:
6.          pass
```

Figure 4.
Pagination.

Figure 5.
Infinite Scrolling.

TripAdvisor presents hotel reviews over multiple separate pages. Once you have successfully scraped the information from the first page of reviews, you may want to move on to the following page to scrape older reviews. This is called pagination (Figure 4; Zhang et al., 2020). We can use the `execute_script` method again to send a JavaScript command to the browser. This method clicks on the "Next page" button, enabling researchers to scrape the same information on the following pages (Listing 9).

Listing 9.
Clicking the Next Page Button.

```
1.  element = driver.find_element_by_xpath('//a
    [contains(@class, "nav next")]')

2.  driver.execute_script("arguments[0].
    click();", element)
```

Reviews on Expedia. One big advantage of Expedia is that all reviews are written by valid customers. That is, as Expedia only allows customers who purchased through their website to write reviews, there is a smaller chance that users will see fake reviews written by non-verified customers on Expedia than on other websites such as TripAdvisor, where anyone can write reviews. Moreover, as Expedia sends out email requests to everyone who makes purchases through their website, the entire review collection process is similar to a survey. Therefore, as long as we know who did not respond to the email request sent by Expedia, we can generalize the study results to the population of those who purchased through Expedia. This may not be possible when using data from TripAdvisor, as the TripAdvisor customers' sample frame is unknown.

On Expedia, users may not need to click the "Next page" button as they do on TripAdvisor. Instead, they must scroll down and load older reviews by clicking on the "More reviews" button shown in Figure 5. This is called infinite scrolling. If there is a significant quantity of reviews written

about a given hotel, users must scroll further down and repeatedly click on the "More reviews" button until all the reviews have loaded. The scraper must mimic this process. Listing 10 opens the number of reviews that you assign with `revnum` (e.g., we set here as 300). This while-loop continues scrolling down until the accumulated number of loaded reviews exceeds `revnum`.

Listing 10.
Scrolling Down.

```
1.  revnum = 300

2.  loadednum = 0

3.  while loadednum < revnum:

4.      more = driver.find_element_by_xpath
        (".//button[contains(@class,'more-
        reviews-button')]")

5.      driver.execute_script ("arguments[0].
        click();", more)

6.      time.sleep(1)

7.      containers = driver.find_elements_
        by_xpath(".//div[contains(@class,
        'uitk-card-separator-bottom')]")

8.      loadednum = len(containers)
```

Reviews on Airbnb. Airbnb is a peer-to-peer marketplace that emerged as a typical example of what is called the sharing economy. Many studies present evidence that Airbnb threatens the traditional accommodation market system (Zervas et al., 2018). This website is most interesting to researchers hoping to better understand customers' behavior in the sharing economy. Due to the peer-to-peer market nature, previous studies argue that there are unique rating behaviors in Airbnb (Ert & Fleischer, 2019). For instance, in addition to guests rating service providers, guests are also evaluated by the host, and these ratings are made publicly available as well. Owing to this dual review process, suppliers tend to receive overwhelmingly high ratings that are not observed under other hotel review systems (Zervas et al., 2018). Despite these unique features of the sharing economy, little is understood about this system. We hope that, by making Airbnb data more accessible, we can encourage other researchers to further explore the sharing economy.

The scraping code for Airbnb is a combination of the codes used for TripAdvisor and Expedia, as the website randomly changes their review display. That is, when the WebDriver visits the supplier's page on Airbnb, the reviews are randomly listed either as pagination or infinite scrolling. Therefore, once the WebDriver opens Airbnb, we need to first determine which scraper should be applied based on the HTML structure. Although there are many different ways to accomplish this, here we build a `check_exists` function that checks whether Airbnb reviews are displayed in pagination or infinite scrolling.

```
1.  def check_exists(self, xpath):
2.      try:
3.          self.find_element_by_xpath
            (xpath)
4.      except NoSuchElementException:
5.          return False
```

After building the function that checks which platform design we have to deal with, we can simply run different scrapers by writing the following if statement.

```
1.  # example of xpath that uniquely
    indicates Infinite scrolling
2.  xpath = './/a[@class="_16i7snfh"]'
3.  if check_exists(driver, xpath)==True:
4.      # Codes for infinite scrolling
5.  else:
6.      # Codes for pagination
```

Reviews on hotel brand sites. Major hotel brands have their own websites where customers can write reviews about their experience, just as they can on TripAdvisor or Expedia. Brands offer up their own reviews in an effort to reduce the need for prospective consumers to visit other travel sites such as TripAdvisor or Expedia. Hotel brands are motivated to do this because reservations made on sites such as TripAdvisor and Expedia are more costly to hotels and because customers who visit these sites may end up booking their stay with another company. It is possible that customers who read and write reviews on hotel brand websites differ from those who use other platforms where alternative hotel options are listed. For instance, these customers may be more loyal to the brand or hoping to have their opinions heard by the hotel manager rather than by other potential customers. Platforms like this, where the user groups differ from other platforms users, generate lots of interesting research opportunities.

Although the scraper should be written differently depending on whether the targeted hotel website uses pagination or ultimate scrolling, the overall process of writing the scraper remains exactly the same regardless of the target website. We include an illustration of Marriott.com in Appendix A (along with complete scripts for other travel sites).

Scraping (Prices) at the Market-Level

While scraping online reviews enable researchers to understand customer Word-of-Mouth (WOM) behaviors, scraping prices gives us insight into the market. This requires no major additional functions or methods beyond the scripts that we introduced previously for scraping reviews. The only difference is that we loop over different hotels within a specific market instead of different reviews. Remember that our scraping code differs by whether the page uses pagination or infinite scrolling. While hotel prices listed on Airbnb and TripAdvisor are displayed using pagination, hotels on Expedia are listed using infinite scrolling. Therefore, utilizing the same approach detailed in code Listing 9, we can build our scraper for Airbnb and TripAdvisor to collect price data and click on the "Next" button to move on to the following pages (see code Listing 11).

Listing 11.
Click the Next Button to Collect all Hotel Prices.

```
1.  next_exists = check_exists(driver, '//a
    [contains(@aria-label, "Next")]')
2.  if next_exists:
3.      element=driver.find_elemet_by_
        xpath('//a[contains(@aria-label,
        "Next")]')
4.      driver.execute_script("arguments[0].
        click();", element)
```

In contrast, for Expedia we design the code to scroll down until there are no additional listings before documenting the hotel prices (see code Listing 11). This approach is equivalent to how we scraped the reviews from Expedia in code Listing 12.

Listing 12.
Scrolling Down to Collect all Hotel Prices.

```
1.  while check_exists(driver, ".//button
    [contains(@data-stid, 'show-more
    -results')]"):
2.      more = driver.find_element_by_xpath
        (".//button[contains(@datastid,
        'show-more-results')]")
```

Table 3.
The Number and the Average Length of Reviews Across Three Platforms During 2018–2020.

	TripAdvisor	Expedia	Hotel Brand
Number of reviews	713	2,089	1,760
Average review length	731.12	119.56	141.63

```
3.      driver.execute_script("arguments[0].
        click();", more)

4.      time.sleep(1)

5.      containers = driver.find_elements_
        by_xpath(".//div[contains(@class,
        'link-container')]")
```

Platform Differences Causing Biases

Although collecting and analyzing online review data broadens our understanding, it is important to mention a few relevant caveats. Unfortunately, many studies conducting research using data from online travel sites ignore the biases that arise when using secondary data. Multiple selection biases exist in scraped data, which can impact a researcher's ability to draw insights about the target population (i.e., all customers who stayed at the hotel). In this section, we present some of the noteworthy differences between hotel review platforms that hospitality researchers should be aware of before collecting and using web scraped data.
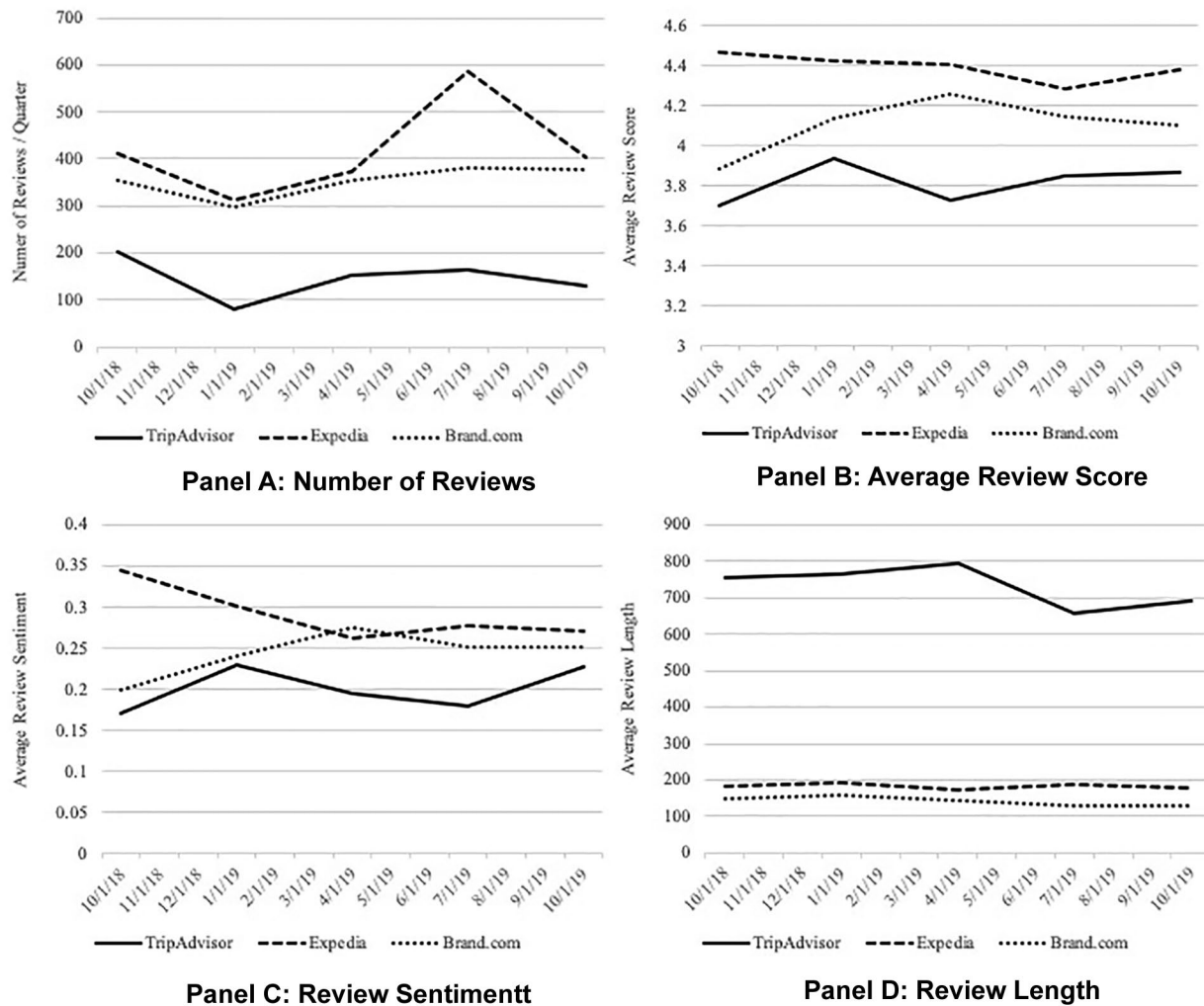
Review Differences Across Platforms

Owing to differences in how reviews are collected and written, collected reviews may vary in terms of number, valence, and content depending on what hotel review platform they were posted on (Litvin & Sobel, 2019). This is true even for reviews that were written about the same hotel and during the same time period. This is true for a couple of reasons. First, every hotel review platform has its own objectives and its own review collection process. To encourage review submission, Expedia sends customers a post-stay email with a link to submit a hotel review. TripAdvisor, however, relies fully on customers' self-motivation to post reviews: unless individual hotels choose to partner with TripAdvisor to invite their verified customers to post reviews, either by sending an email or through their online reputation management firms (e.g., ReviewPro, Revinate, and Medallia). These differences in the review collection process among these platforms yield systematic differences in terms of the review scores and their content. Second, customers may perceive different platforms as having different audiences. Therefore, different types of customers may prefer different platforms depending on who they hope to reach with their review. While customers may perceive distributors' websites (i.e., a hotel's own website or Expedia) as a suitable channel for contacting hotel managers, TripAdvisor might be perceived as an appropriate method for reaching other potential customers. As a result, the dominant valence and topics may differ across platforms.

Several noteworthy patterns are frequently observed across travel platforms. First, a major difference between review platforms is the number of reviews per hotel. To illustrate how a given hotel's quality might be evaluated differently by each of the three platforms, we scraped the reviews of the New York Marriott Marquis, which has a large number of reviews on TripAdvisor, Expedia, and Marriott's own website. For the purposes of this demonstration, we utilized reviews written between October 2018 and December 2019. Table 3 demonstrates that significantly more reviews were posted on Expedia and the hotel's own website than on TripAdvisor. This pattern is particularly interesting because anyone can post a review on TripAdvisor, whereas only verified guests can post reviews on OTAs and hotel brand sites. A major driver in creating this disparity in the number of reviews across the platforms stems from how the reviews are collected. Expedia and hotels encourage customers to post reviews online by sending post-stay feedback invitation emails. This pattern holds across different time periods, as shown in Figure 6, where we plot review characteristics (number, score, sentiment, and length) on a quarterly basis for 2018 and 2019. The average review length is significantly longer on TripAdvisor than on the other platforms. Self-motivated (i.e., non-email prompted) reviewers are more likely to put greater effort into writing reviews, given that they have already undertaken the effort of opening TripAdvisor and posting a review. Therefore, review length (calculated in number of words) is greatest on TripAdvisor, where most reviews are self-motivated. After eliminating stop words, which refer to the words that are filtered out due to their extremely high or low appearances, we counted the words that appeared on each platform. As is shown in Table 4, the most frequently used words are slightly different depending on the platform. Unlike Expedia and TripAdvisor, where the most frequent word is "room," the hotel's own site reviewers are more focused on the hotel location.

Finally, the distribution of ratings between the two platforms has different patterns, as is shown in Figure 7. The proportion of relatively negative ratings is higher on TripAdvisor than on Expedia or the brand site. Given the cost or effort of posting a review on TripAdvisor, customers may be more likely to post if their expectations have not been met (or have been exceeded), and as a result these

Panel A: Number of Reviews

Panel B: Average Review Score

Panel C: Review Sentimentt

Panel D: Review Length

Figure 6.
Differences in Reviews (Number, Score, Sentiment, and Length) by Platform.

Table 4.
Top 10 Most Frequently Used Words for the Same Hotel's Reviews by Platforms.

	TripAdvisor	Expedia	Hotel Brand
1	room (1,346)	room (754)	location (678)
2	time (1,035)	location (691)	room (654)
3	stay (659)	time (482)	stay (644)
4	square (536)	staff (400)	time (562)
5	location (463)	square (334)	staff (488)
6	get (367)	stay (288)	square (352)
7	service (360)	leave (287)	service (295)
8	one (349)	review (282)	excellent (228)
9	staff (336)	comments (281)	good (217)
10	floor (331)	traveler (281)	clean (210)

reviews may be more extreme (and perhaps nonrepresentative; Anderson et al., 1994). In contrast, reviews that were more easily posted (due to the email invitation) are less likely to be affected by the different posting motivation across different ratings. As a result, reviewers that would otherwise have relatively low posting intention (i.e., positive ratings) are able to easily submit their reviews, potentially resulting in higher average ratings at sites with email encouraged/prompted reviews, for example, OTAs and brand sites (Han & Anderson, 2020).

Price Differences Across Platforms

Next to reviews, prices are among the most interesting data types available across different travel platforms. Just as
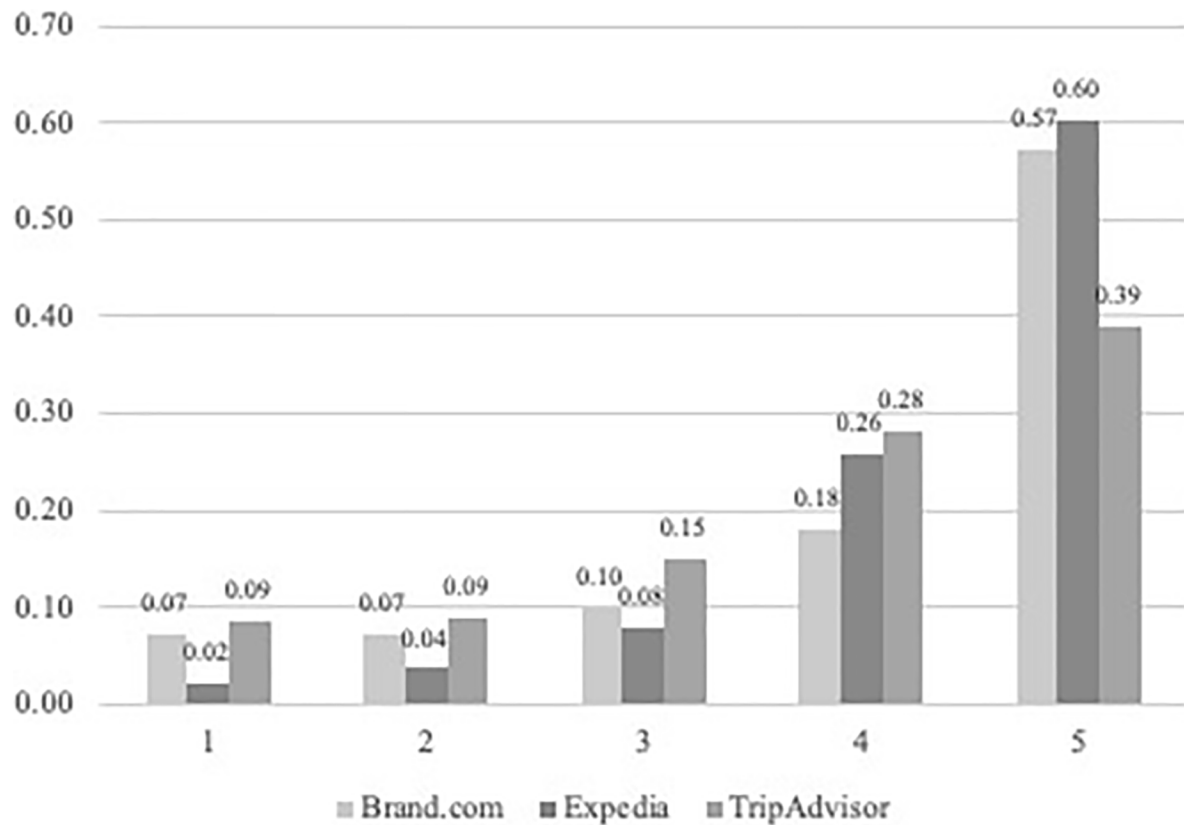
Figure 7.
Distribution of Review Scores by Platform.

Figure 8.
Listing of a Sold-Out Property in Expedia.

with reviews, different platforms have different objectives and different capabilities when presenting prices. As a result, the way in which prices are displayed and communicated across platforms varies, which may influence customers' purchasing decisions. For example, Expedia displays all properties, including those that are sold-out or have limited availability, as is shown in Figure 8. In contrast, other travel platforms, such as Airbnb, only list properties that are available and have price data to be displayed. Customers may behave differently as a result of these differences in information display. Accordingly, it is likely that customers behave differently depending on which travel platform they choose (Park & Jang, 2018). Therefore, assuming that all platforms have the same market structure ignores these differences in information stimuli that influence customer behavior.

OTAs are also considerably more versed in merchandising when displaying price information. Merchandising by OTAs may include such actions as strike-through pricing or scarcity messaging, which attempt to increase conversion

Figure 9.
Different Price Offer to the Members.

rates. Similarly, rate parity has received considerable attention recently. While OTAs want to display prices that can compete with those posted on supplier websites, hotels are often motivated to offer better prices to customers who book directly to avoid OTA commissions. Closed user group (CUG) or membership selling is a common method of offering lower prices to a subset of consumers. In CUG situations, the consumer must login (either at the OTA or hotel site) to access better prices. Figure 9 illustrates strikethrough prices and the resulting member rates available to a typical CUG. By definition, CUG discounted prices are only available to those who have logged in; the majority of customers will never see this price disparity. Researchers who analyze the online hotel market using prices scraped from travel websites must understand how these distribution channels influence each other.

Implications

The academic implications of web scraping are twofold: exploratory and confirmatory. In general, web scraped data are suitable for exploratory research where the research question has not been examined in detail. The aim of this research area is to understand the general pattern of the customers in question and find preliminary evidence that warrants a more detailed study. For example, Danescu-Niculescu-Mizil et al. (2013) show that users in online beer communities follow a two-stage life cycle in terms of the language they use: the innovative learning phase and the conservative phase. Wu and Huberman (2010) found that later online product review ratings tend to vary considerably from earlier ones, making overall review ratings less extreme. Although these studies demonstrate interesting behavioral patterns and

are valuable in their own right, they still require further testing in an experimental setting.

Web scraped data are also often used in confirmatory studies that test hypotheses from previous studies (Landers et al., 2016; Marres & Weltevrede, 2013). If this hypothesis testing investigates whether one of the variables scraped from the website is an exogenous variable that varies independently of an error term, this study is considered as an experiment (Harrison & List, 2004). Web scraped data enable researchers to conduct experiments in a real environmental context and conduct research without informing research subjects that they are taking part in an experiment. This research design is referred to as a natural experiment. The advantage of a natural experiment is that it accounts for the realistic environment that real customers encounter. Therefore, an ideal natural experiment not only increases external validity but also does so when internal validity is insufficient (Harrison & List, 2004). For example, Han and Anderson (2020) take advantage of a unique characteristic of TripAdvisor, the fact that a portion of its reviewers post after being prompted to do so by hotel managers. By comparing regular self-motivated reviews and prompted reviews, the authors test whether satisfied or dissatisfied customers are more motivated to post online reviews. Web scraped data can also be combined with data from other sources in confirmatory research. Xie et al. (2014) combined TripAdvisor's review data with archival data regarding hotel revenue per available room (RevPAR) matched to the Texas Comptroller's Office database to investigate the impact of various review website attributes on hotel performance.

Managers in the hospitality industry can benefit from web scraping, too. It is well known that customers heavily rely on online reviews before making a purchasing decision

(Brown et al., 2007; Chevalier & Mayzlin, 2006). Therefore, it is important to understand how customers evaluate services. Although there are multiple online reputation management companies that do this job on behalf of firms, hotels can save money by web scraping themselves. For example, firms can web scrape major online review websites in the industry periodically and automate this process, which is referred to as web crawling (Massimino, 2016). Without spending any extra money, firms can obtain a summary of each online review website. Based on these summaries, managers can decide which review websites they should prioritize and invest to maximize the number of reviews they receive.

Ethical Concerns

As legislation on web scraping varies from country to country, researchers should look into local legislation. In this section, we discuss the legal and ethical issues mainly in the U.S. context. On September 9, 2019, the U.S. Supreme Court legalized web scraping in situations where the scraped information is designed to be publicly accessible. The court defined public information as data that are neither available for purchase nor hidden behind a password-protected authentication system. The logic behind the court's decision was that, legally, web scraping is no different than browsing in terms of what data are being requested from a website. However, web scraping information that is accessible exclusively to the members and requires logging in is illegal, as this behavior explicitly violates the terms of service (ToS).

It is also noteworthy that web scraping copyrighted data and re-using them for commercial purposes would be considered illegal. For example, web scraping video contents from YouTube and re-posting them on ones' own website could be illegal as videos are copyrighted.

However, illegally sharing data is not likely a matter of concern for the majority of our target audience: researchers who are interested in using web scraping for academic purposes. In addition, web scraping is a relatively new data collection method in academia, and therefore the law is still evolving (Hillen, 2019). Therefore, researchers must take into account that it is always possible that current laws regarding web scraping will change and that they may need to seek professional legal advice before web scraping.

In addition to legal issues, researchers should also consider ethical constraints when collecting data online for academic purposes. As previous studies argue, legality does not necessarily mean that data usage is entirely ethical (Massimino, 2016). Using online data in research is relatively new in comparison to other data sources. Therefore, its current legality suggests a need for further research regarding the ethicality of web scraped data and the safety of the web scraping practice. By the same token, although web scraping qualifies for Institutional Review Boards (IRBs)' review exemptions in most of the cases (Massimino, 2016), researchers need to be conscientious of any societal entities that may be impacted by web scraping. A common ethical concern regarding web scraping is related to the problem of sending too many requests to the host over a short span of time. A typical web scraper involves querying a website repeatedly. If overused, this practice can prevent others from accessing the website. A web scraper that is written for the purpose of collecting multiple online reviews or hotel prices sends requests to the web server that is hosting the site whenever it opens a new page. While the requests of a human user are usually within a manageable range, a web scraper that makes speedy and bulky automated requests can easily exceed the bandwidth threshold of the host and make the server unresponsive (Massimino, 2016). When the web scraper hits the server with frequent requests, a host may issue a warning or may respond with useless content if web scraping behavior is detected.

Therefore, we recommend inserting a random delay between individual requests, such as limiting requests to three per second (Massimino, 2016). For example, Landers et al. (2016) executed a 2-s delay between each web page request to avoid overburdening the host's server. In addition, scraping during off-peak hours can help reduce the load on the host and increase the speed of the scraping process. Finally, a good practice in web scraping is to carefully read the robots exclusion protocol (REP), which are standardized instructions on whether certain user agents can scrape parts of a website. Typically, this information can be found in the admin page of a website.

Summary

A growing number of customers not only obtain travel information online but also make transactions over the internet. Studies indicate that the internet (as opposed to the offline, voice, or travel agent distribution channels) has become the dominant distribution channel in terms of travel reservations (Park, 2009). Therefore, the importance of studying online customer behaviors cannot be overemphasized. However, there is a strong tendency among hospitality researchers to rely on traditional data collection methods, which are limited in terms of what research questions they can answer as well as the generalizability of their insights. Our work provides a simple method of scraping online reviews and price data using the Python language. As our goal is to make hospitality researchers more comfortable collecting online data, we focus on how the scraping process functions on major travel websites. Although not a comprehensive introduction into Python, we introduce the essential elements necessary to handle and scrape interactive

hospitality platforms such that they can augment readily available introductory Python resources.

Although web scraped data introduce incredible opportunities to hospitality researchers, there are important aspects that must be accounted for when scraping travel websites. As every platform has unique characteristics and purposes, each platform attracts different users, which in turn forms a different market. While ignoring these platform-specific aspects may induce biases in the researchers' analyses, properly making use of these challenges may make platform differences a unique natural experiment setting for exploring new opportunities. At the same time, embracing these differences is what provides for fruitful research. For instance, the social influence effect, which refers to the effect of previous reviews on future reviews, impacts most online review platforms due to the nature of being able to see previous reviewers' opinions. This effect induces potential biases that are not of concern in traditional survey research as there is no chance that survey participants will see other participants' opinions before answering the survey questions. Despite the possibility of inducing social influence bias, Askalidis et al. (2017) overcome this challenge by utilizing reviews written by retailer-prompted reviewers who were invited to contribute their opinions using a separate web page where there is a lesser chance of seeing previous opinions. Going further, they compare this group of reviews to the regular, organic reviews using the difference-in-difference method, and make use of the challenge to identify the social influence effect in the online communities. Another example of turning the challenge into opportunity is the study of Wang and Chaudhry (2018). Many online hotel review platforms allow managers to respond to customer reviews, which may influence future reviewers' opinions. While the managerial response could present a challenge for researchers who want to understand the unbiased opinions of the reviewers, Wang and Chaudhry (2018) identified the managerial response effect by comparing ratings from online review platforms where managerial responses are visible with ratings from platforms where managerial responses are not made visible.

Our article has significant implications for hospitality researchers who hope to better understand the online travel marketplace. We outline simple methods that enable hospitality researchers to collect incredibly useful secondary data that they could not have obtained by relying on traditional data collection methods alone. Although traditional data collection methods are still valuable and, in many cases, cannot be replaced by new methods, online customer behaviors are hardly replicable in offline research design. Even if the purpose of the research is to make a causal inference that can only be tested in a strict lab setting, confirming this effect in the real online marketplace adds value to the study in terms of external validity.

ORCID iDs

Saram Han [iD] https://orcid.org/0000-0001-9077-9107
Christopher K. Anderson [iD] https://orcid.org/0000-0002-7103-8351

Supplemental Material

Supplemental material for this article is available online.

References

Anderson, E. W., Fornell, C., & Lehmann, D. R. (1994). Customer satisfaction, market share, and profitability: Findings from Sweden. *Journal of Marketing*, *58*(3), 53–66.

Askalidis, G., Kim, S. J., & Malthouse, E. C. (2017). Understanding and overcoming biases in online review systems. *Decision Support Systems*, *97*, 23–30.

Brown, J., Broderick, A. J., & Lee, N. (2007). Word of mouth communication within online communities: Conceptualizing the online social network. *Journal of Interactive Marketing*, *21*(3), 2–20. https://doi.org/10.1002/dir.20082

Chevalier, J. A., Dover, Y., & Mayzlin, D. (2018). Channels of impact: User reviews when quality is dynamic and managers respond. *Marketing Science*, *37*, 685–853.

Chevalier, J. A., & Mayzlin, D. (2006). The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research*, *43*(3), 345–354. https://doi.org/10.1509/jmkr.43.3.345

Choi, S., Mattila, A. S., Van Hoof, H. B., & Quadri-Felitti, D. (2017). The role of power and incentives in inducing fake reviews in the tourism industry. *Journal of Travel Research*, *56*(8), 975–987.

Danescu-Niculescu-Mizil, C., West, R., Jurafsky, D., Leskovec, J., & Potts, C. (2013). No country for old members. *Proceedings of the 22nd International Conference on World Wide Web - WWW '13, Rio de Janeiro, Brazil* (pp. 307–318). Association for Computing Machinery. https://doi.org/10.1145/2488388.2488416

Downey, A. (2014). *Think Python: How to think like a computer scientist*. Green Tea Press.

Ert, E., & Fleischer, A. (2019). The evolution of trust in Airbnb: A case of home rental. *Annals of Tourism Research*, *75*, 279–287.

Gao, B., Hu, N., & Bose, I. (2017). Follow the herd or be myself? An analysis of consistency in behavior of reviewers and helpfulness of their reviews. *Decision Support Systems*, *95*, 1–11.

Han, S., & Anderson, C. K. (2020). Customer motivation and response bias in online reviews. *Cornell Hospitality*

Quarterly, *61*, 142–153. https://doi.org/10.1177/19389655 20902012

Harrison, G. W., & List, J. A. (2004). Field experiments. *Journal of Economic Literature*, *42*(4), 1009–1055.

Hillen, J. (2019). Web scraping for food price research. *British Food Journal*, *121*(12), 3350–3361. https://doi.org/10.1108/ BFJ-02-2019-0081

Kupor, D., & Tormala, Z. (2018). When moderation fosters persuasion: The persuasive power of deviatory reviews. *Journal of Consumer Research*, *45*, 490–510.

Landers, R. N., Brusso, R. C., Cavanaugh, K. J., & Collmus, A. B. (2016). A primer on theory-driven web scraping: Automatic extraction of big data from the internet for use in psychological research. *Psychological Methods*, *21*(4), 475–492.

Litvin, S. W., & Sobel, R. N. (2019). Organic versus solicited hotel TripAdvisor reviews: Measuring their respective characteristics. *Cornell Hospitality Quarterly*, *60*(4), 370–377.

Marres, N., & Weltevrede, E. (2013). Scraping the social?: Issues in live social research. *Journal of Cultural Economy*, *6*(3), 313–335.

Massimino, B. (2016). Accessing online data: Web-crawling and information-scraping techniques to automate the assembly of research data. *Journal of Business Logistics*, *37*(1), 34–42.

Mayzlin, D., Dover, Y., & Chevalier, J. (2014). Promotional reviews: An empirical investigation of online review manipulation. *American Economic Review*, *104*(8), 2421–2455.

Min, H., Lim, Y., & Magnini, V. P. (2015). Factors affecting customer satisfaction in responses to negative online hotel reviews: The impact of empathy, paraphrasing, and speed. *Cornell Hospitality Quarterly*, *56*(2), 223–231.

Park, J. (2009). Consumers' travel website transferring behaviour: Analysis using clickstream data-time, frequency, and spending. *Service Industries Journal*, *29*(10), 1451–1463.

Park, J. Y., & Jang, S. C. S. (2018). The impact of sold-out information on tourist choice decisions. *Journal of Travel and Tourism Marketing*, *35*(5), 622–632.

Schahn, J., & Holzer, E. (1990). Studies of individual environmental concern. *Environment and Behavior*, *22*(6), 767–786.

Shin, H., Perdue, R. R., & Pandelaere, M. (2019). Managing customer reviews for value co-creation: An empowerment theory perspective. *Journal of Travel Research*, *59*, 792–810. https:// doi.org/10.1177/0047287519867138

vanden Broucke, S., & Baesens, B. (2018). *Practical web scraping for data science*. Apress.

Viglia, G., & Dolnicar, S. (2020). A review of experiments in tourism and hospitality. *Annals of Tourism Research*, *80*, 102858.

Wang, Y., & Chaudhry, A. (2018). When and how managers' responses to online reviews affect subsequent reviews. *Journal of Marketing Research*, *55*(2), 163–177.

Wu, F., & Huberman, B. A. (2010). Opinion formation under costly expression. *ACM Transactions on Intelligent Systems and Technology*, *1*(1), 1–13. https://doi.org/10.1145/1858948 .1858953

Wu, L., Mattila, A. S., Wang, C. Y., & Hanks, L. (2015). The impact of power on service customers' willingness to post online reviews. *Journal of Service Research*, *19*(2), 224–238.

Xie, K. L., Zhang, Z., & Zhang, Z. (2014). The business value of online consumer reviews and management response to hotel performance. *International Journal of Hospitality Management*, *43*, 1–12. https://doi.org/10.1016/j.ijhm.2014.07.007

Zervas, G., Proserpio, D., & Byers, J. (2018). A first look at online reputation on Airbnb, where every stay is above average. *SSRN Electronic Journal*. http://dx.doi.org/10.2139/ssrn .2554500

Zhang, Y., Liu, L., & Ho, S. Y. (2020). How do interruptions affect user contributions on social commerce? *Information Systems Journal*, *30*(3), 535–565.

Author Biographies

Saram Han is an assistant professor at the College of Business and Technology at Seoul National University of Science and Technology. He received his PhD from the Cornell School of Hotel Administration in the Cornell SC Johnson College of Business. His research interests include digital marketing, online reviews, service marketing, and marketing analytics. He earned a BBA in Tourism Management from Kyung-hee University, Seoul, Korea, and an M.S. degree from the Michigan Program in Survey Methodology, University of Michigan.

Christopher K. Anderson is a professor at Cornell SC Johnson College of Business, School of Hotel Administration. He earned his BSc/MSc in engineering from the University of Guelph, and his MBA/PhD from the University of Western Ontario, Richard Ivey School of Business. He teaches and conducts research in data analytics, pricing, distribution, and revenue management.