Individual Integrative Assignment

Data management for good: the social and economic impact of Airbnb

Khanh Chu Nam (554897)

BM02BAM: Data Management & Ethics

Lecturer: Dr. Anna Priante

Word count: 5426 words

October 9th, 2023

I. Problem statement & question generation

At the time of writing, Paris has been, and will be, greeting a prolific amount of guests throughout the upcoming twelve months. In addition to the usual tourist scenery, the city is also host for the Rugby World Cup (September-October 2023) and the 2024 Olympics – Paralympics Games (July – September 2024). At the same time, holidays, recurring and local events – like the Paris Fashion Week or Tour de France – are also happening in the city. Because of this, Paris is anticipating a sharp increase in short-stays for the upcoming month, especially around the next summer period (Airbnb, 2023). In fact, locals are already trying to capitalize on these events, going overboard with the price increase – ranging from a forecasted 85% (Lloyd, 2023) to even over 200% (Gadeva, 2023).

Seeing the magnitude of accommodation price inflation, the government of France has already signed a charter demanding Peer-to-peer (P2P) accommodation platforms like Airbnb to notify users of excessively high prices (Lloyd, 2023). For government, a property is putting an unfair price if its price is 'much higher' than properties of the same size in the same date. A notification will then be shown when visitors access property information, preventing them from being "ripped off in France" (Lloyd, 2023).

While government intervention is justifiable, the criteria they use to assess accommodations are not. Size is not the only explanatory variable for the price, and same-date estimation encourages price escalation - where hosts jointly increase price as a response to each other, causing overall high price for the market – that is detrimental to unsuspecting tourists. A more intuitive, history-based price alert for Paris needs to be implemented, so that tourists and hosts with odd size properties do not simply fall victim to the alert. For this price alert to work, the municipality of Paris needs to understand pricing trends and affecting factors in the city – which is where this project comes in.

Within the scope of this project (assignment), we consider accommodation (hereby referred to as 'listings') on Airbnb. Compared to competitors such as Booking.com, Airbnb in France is one of the most used travel apps (Similarweb, n.d.) and covers a wide variety of accommodation apps from shared, private to even commercialize. Information about listings on Airbnb is also publicly available and accessible through the project "Inside Airbnb" – a project dedicated to "control the role of renting residential homes to tourists" (Inside Airbnb, 2023). Metadata from "Inside Airbnb" can also be replicated/scraped on other P2P platforms – opening doors to expand the project scope in the future.

To implement an intuitive Airbnb price alert, the municipality need past information about pricing, in relation to accommodation (type & neighborhood) and host (status and credibility); after adjusting for markup during holidays & weekends, the municipality can work out the expected average price (per person) in the past year. From here, they can either calculate a hedonic function or implement fixed pricing alert based on a specific criterion, adjusted to 2024 context. This project supports the Paris municipality in the first step, in which it aims to:

Understand the (listed) asking average price per person of short-stay Airbnb listings in Paris, and how selected factors of listings - accommodation, host, markup - individually affect this price in the past year.

In order to solve this problem, the following questions will be answered in the project:

In the past year, on Airbnb…

- What was the average asking price (per person, per night) of active short-stay listings in Paris?
- What was the average price of Paris listings, during weekends/holidays/normal days?
    - Consequently, what is the expected markup percentage for special days for price?
- What is the difference in average price of Paris listings, based on neighborhood the listing is in?
- What is the difference in average price of Paris listings, based on room type of the listing?
- What is the difference in average price of Paris listings between superhosts and normal hosts?
- What is the difference in average price of Paris listings between professional hosts (having more than one listing) and casual hosts?
- For top 5% of listings in terms of quality (high review score from many reviews), what is the difference in average price, compared to the rest of listings?

The final question is meant to investigate whether an upward skew of average price exists due to high-quality listings offered at a worthy-but-significantly higher price. Other questions help to inform understanding about average price per person of Paris listings through 4 factors (Neighborhood, room type, superhost status - an exclusive status on Airbnb, host experience), while taking account of price markup during holidays. While there can be many factors determining the price of an Airbnb listing – going up as high as 150 in previous studies (Moreno-Izquierdo et al., 2018) – this project based the selection on a subset used by Gibbs et al. (2017), which is compact yet good explanation of variance, and available with the data scrape from Inside Airbnb project.

Note that this project adheres to Data Policies of InsideAirbnb (2023a) and GDPR. Data is downloaded from the website on September 12, 2023 and used throughout the project. This project makes no use of personal data and provides best effort to cover them – regardless of whether they can be traced back to an individual, or whether the information is public. Data retention follows the policies of Data Management and Ethics course at Erasmus University Rotterdam, of which this project is within scope of.

II. Design & Organize

Specifically in this project, the dataset used to analyze Airbnb listing price is the 09 October 2022 version (Inside Airbnb, 2023). This dataset provides (listed) price of listings in Paris up until 365 days in the future from the time of scraping, meaning that the analyzed period (September 2022 – September 2023) will not overlap with Rugby World Cup – a potential confounding element for price.

Table 1 introduces variables associated with entities of interest. As mentioned in part I, variable selection is based on study by Gibbs et al. (2017), available data from InsideAirbnb, and common knowledge. Newly created/transformed variables to answer the research questions are highlighted in blue in the ERD and the table below. The method in which they were created & rationale to use in place of available data is also provided. Where not specified, source of variable description is taken from the latest version of Data Dictionary (Inside Airbnb Data Dictionary, n.d.)

Table 1

Description of variables included in analysis of project

Variable name	Description	Source	Notes
redacted_listing_id	Airbnb's unique identifier for the listing, anonymized	See section III	Transformed to follow 'data minimization' & 'data protection by design' guidelines of GDPR (IT Governance Privacy Team, 2020)
redacted_host_id	Airbnb's unique identifier for the host/user, anonymized		
neighbourhood_cleansed	The neighbourhood as geocoded using the latitude and longitude against neighborhoods as defined by open or public digital shapefiles.		
room_type	4 values to indicate: Entire room, Private room, Shared room, Hotel		Data dictionary is outdated
accommodates	The maximum capacity of the listing		
review_scores_rating	Average rating for listings from visitors, in terms of overall quality	Definition not specified in data dictionary, can be deduced by common sense	
number_of_reviews	The number of reviews the listing has		Criteria supplementing *reiew_scores_rating; See section V for rationale*
price_per_person	The (asking) price for listings in upcoming 365 days [September 2022 – September 2023] as indicated by hosts, for each person in the booking	Taken as average of available prices in *(calendar).price*, divided by *accommodates*	Original price variable does not reflect the markup price behavior & accommodation size

calendar_last_scraped	The date in which the price calendar was scraped by Inside Airbnb		For data documentation purposes (CESSDA Training Team (2020)
host_is_superhost	A boolean indicating whether host was granted superhost status from Airbnb – for offering top experience (Airbnb, n.d)		An indicator similar to perceived popularity in P2P accommodation (Ferguson & Ryan, 2018).
host_total_listings_count	The number of listings the host has (per Airbnb calculations)		
(calendar).date	The date in the listing's calendar		
(calendar).price	The price listed for the day		
special_day	The date that is a weekend (Saturday/Sunday), a national holiday of France, or a weekend immediately after a national holiday	National holidays are taken from (PublicHolidays.fr, 2022, 2023)	See also section IV for rationale

Some variables are used in the data cleaning process, but are not used in the analysis process. These variables are described in Table 2 below. In accordance with data minimization principle of GDPR (IT Governance Privacy Team, 2020), these variables are not included in the analysis database.

Table 2

Description of variables not analyzed, but used in this project

Variable name	Description	Source	Notes
minimum_nights	minimum number of night stay for the listing (calendar rules may be different)		To focus on short-stay listings, listings with minimum_nights ≥ 30 are dropped
availability_365	The availability of the listing 365 days in the future as determined by the calendar. Note a listing may not be available because it has been booked by a guest or blocked by the host.		Listings with availability_365 = 0 are dropped, due to being unrealistic (Qiu et al., 2019)

The variables are taken from *listings.csv* and *calendar.csv* file of Inside Airbnb, and have been normalized appropriately. Among them, variables *calendar_last_scraped* and *date* are considered to contain atomic values. The columns might have separable information about day, month and year, but together they signify a calendar day – which serves the purpose of analyzing price based on calendar - and are treated like one. Furthermore, from Table 1, the following set of functional dependencies hold:

redacted_listing_id-> redacted_host_id, neighbourhood_cleansed, accommodates, review_scores_rating, number_of_reviews, price_per_person, calendar_last_scraped

redacted_host_id -> host_is_superhost, host_total_listings_count

redacted_listing_id, date -> price

date -> special_day

Each variables in a functional dependency has been formed into a new table. Since this set of functional dependency is also the canonical cover set for the dataset, no partial dependency or transitive dependency can be found in the resulting tables. Thus, the relation satisfies the third normal form.

Figure 1 presents the physical Entity Relationship Diagram of this project:

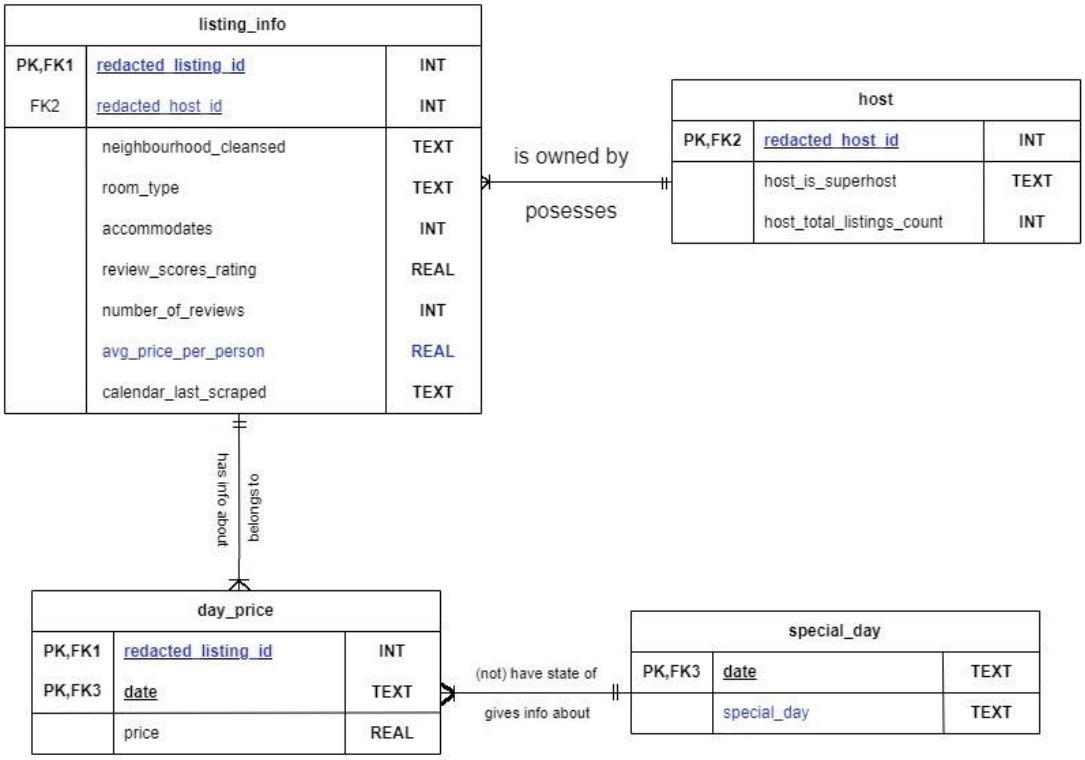

Figure 1: ERD of this project

Three entities are of interest in this project: The listing (including price – primary variable), the calendar and the host. A listing is owned by one and only one host, while a host can own from one-to-many listings. A listing also has one to many days with information about price, while a particular price has to belong to a specific property on a specific day. For the day price table, a special day reference entity is associated with it, to indicate whether the date (and the price associated with it) is in the weekend/holiday or not. The price of a property on a given day only has one state (weekend/holiday or not), but the determined state of a day determines the property of many prices from different listings.

III. Data processing

The data processing closely follows the steps outlined in the guide (Priante, 2023), with slight deviations where convenient.

The files *listings.csv* and *calendar.csv* from 09 September 2022 are imported into the database. From those tables, columns *calendar.price*, *review_scores_rating* are adjusted from data type TEXT to REAL. While not necessary, columns *adjusted_price*, 5 other *review_scores_ (rating, accuracy, cleaniness, checkin, communication, location, value)* also have their datatype changed for bookeeping purposes.

Also for tracking puposes, column *id* from listings table is renamed to *listing_id*. Otherwise, all variables are in English memonic naming convention, outlined by CESSDA Training Team (2020, p. 40).

A quick run-through of the tables and row count reveals that there are 61365 properties in the data set, with over 2 million records of their day price. By counting the potential primary key *listing_id, host_id*, composite of *listing_id* and *(calendar.)price* that is more than 1, a query is performed to find duplicate records in the two tables. There were no results returning, indicating that there are no duplicates in the dataset.

Handling missing data and filtering out inconsistent (numeric) rows are done simultaneously at this stage. In the *listings* table, a null value check was conducted for every variable included in the ERD. The following variables have null values:

- host_is_superhost (44 listings)
- host_total_listings_count (7 listings)
- review_scores_rating (11806 listings)

Since null values in *host_is_superhost* and *host_total_listings_count* cannot be recovered within the dataset - affect the query and outcome of research questions, rows having them are removed from the dataset. *review_scores_rating* is also not recovered, but data examination reveals that this is due to these listings not having any reviews yet. Removing listings due to null value in this variable causes substantial loss when understanding the asking price of Paris hosts in the past year. Keeping them instead helps the related query (of top 5% active listings) to be more accurate with real-world, while not producing unwanted outcomes. Therefore, the null values in *review_scores_rating* is kept, and related query using it will ignore null values.

Afterwards, rows in *listings* table with *minimum_nights* > 30, *availability_365* = 0 are removed as well. This ensures realistic, active short-stay rentals are in consideration for analysis – 24798 and 3818 more rows are deleted from the table. For *review_scores_rating*, those with rating < 1 and > 5 are removed, since rating bounds on Airbnb website is [1,5]; this affects 34 listings. Processing other numeric columns of *number of reviews, accommodates, host_total_listings_count* return no other

inconsistent row. Finally, in the *calendar* table, any row having *listing_id* value not remaining in *listings* table is removed. A previous check has already found no missing value in the *calendar* table.

An outlier detection should be conducted now for the numerical variables. However, given the ERD in section II, the only variable that might have outliers adversely affecting analysis is the newly computed *price_per_person*. Since the variable transformation is done at a later stage (for documentation purposes), outliers processing of the dataset is also pushed back.

For checking potential spelling mistakes in *neighbourhood_cleansed*, *room_type* and *host_is superhost*, a query for distinct values in respective columns are used. All values are within possible values of neighbourhood_cleansed (20 values), room_type (4 values) and host_is_superhost(2 values) – based on references from the InsideAirbnb website (2023b).

In the dataset, price columns have a $ sign before them. Not only do they need to be dropped, but they are also inconsistent with real life – as France use €. In the dataset the $ sign have been dropped from the following columns: *calendar.price*; *calendar.adjusted_price* & *listings.price* (as precautionary measure). For this project's analysis, any price-related numbers are treated as in euro currency. Some listings that has price > 1000 euros has a comma in the thousands to signify – this is also dropped.

A quick, final check for multiple values within one column for every variable in the ERD (by checking if a comma presents) revealed no such instances. The dataset has been thoroughly cleaned, and the database can now be implemented – as described in section IV below.

IV. Database implementation

Data is now clean, but the variables used in the ERD are not: host_id and listing_id are not masked into their redacted version, price_per_person has yet to be calculated (and checked for outliers). The implementation stage also documents their creation (alongside the query log).

As host_id and listing_id can be misused to track back into personal data, they need to be transformed when used in ERD. I considered four approaches: Masking with asterisks, hashing, randomize and auto increment. Using asterisks poses several problems: The length of host_id and listing_id vary from 3 to over 10 letters, and there are no meaningful way to put asterisks in without either revealing hints for the method to reconstruct real id, or create duplicate ids. Hashing is not an option in DB Browser. Randomizing and autoincrementing work esentially the same way (both require to transform in tables where *host_id* and *listing_id* are unique), but autoincrementing only needs to be implemented in the schema without additional queries, so I utilized that.

The tables are created in the following order: host -> listing_info -> day_price -> special_day. Table *host* is where the transform of *host_id* into *redacted_host_id* take place – with *listing_info* also using the variable – and is created first. *listing_info* is where *listing_id* is transformed into *redacted_listing_id* that *day_price* also uses. *special_day* uses the day from *day_price* and is created last. Primary key declaration is done in-line, except for the composite key in *day_price* that has a separate line for that.

Since the redacted ids need to be traced back to the original ids, tables with redacted ids also have column for original ids at first. When importing data from dataset to ERD tables, redacted id columns are left blank (so autoincrementation of DB Browser can work). Afterwards, the *listings* table has columns to match redacted ids – through the original ids in the ERD tables. I did not do this for *calendar* table, since it has over 1 million rows, update query cannot uses index, and query to match redacted ids will take too much processing time; this cause a problem where *day_price* needs its data from *calendar*, but *calendar* does not have *redacted_host_id*. To circumvent this, I created a temporary relation with *calendar* left joining newly created *listing_info*, essentially matching *redacted_listing_id* in *listing_info* to *calendar*. The temporary relation now has enough data to populate day_price.

Populating *special_day* table also requires effort, since information is inherent within the dataset or DB Browser. First, all distinct dates are pulled from *day_price* dataset, and assigned a label ‘Weekend’ or ‘Normal day’ based on the strftime function. Small note that due to different calendar scrape dates, there are in total 371 days reflected in the dataset. Secondly, dates that are national holidays, as indicated by the website Publicholidays.fr (2022,2023) have their labels manually set to “Holiday”. Finally, for national holidays that are on a Friday or Monday, the weekend after or before

them is also set to holiday – based on a notion that when you have combined days off due to weekend and holiday, you don't consider them separately but as one long holiday. For the last two steps of this process, there exists a query to adjust automatically (with or without separating the column into smaller atomic values), but given the complexity of the query I opt for manually inserting eligible dates instead.

With all tables populated, there is still *price_per_person* value that needs to be calculated – after being delayed to this phase. Values to calculate *price_per_person* are in *day_price*, while the variable is in *listing_info*. Therefore, I created a temporary relation that has the redacted_id and calculated price_per_person first – then match redacted id with the *listing_info* table. This price is calculated as an average of all prices in the available *calendar*, divided by *accommodates* value, for a specific listing. This number has been rounded up to two decimals, and due to the preprocessing in section 3, is not null in any row of the table.

An outlier detection is conducted on this newly created *price_per_person* variable, using standard deviation (SD). As a rule of thumb, 99.7% of values fall within 3 standard deviation of the mean. Since a function to calculate SD is not in DB Browser, three queries (and hand calculations) are done, using the formula:

$$SD = \sqrt{\frac{\sum(x_i - \bar{x})^2}{N}}$$

Before removing the outliers, I have examined their day price in order to understand the reason behind excessive high price. This is either due to the host setting an unrealistic high price for a period in the calendar (tens of thousands per night) instead of blocking booking, or Airbnb place the numbers as placeholders. In both cases, the values are skewing the dataset and thus removed. No values are within the lower bounds of SD (which is already unrealistic as it is in the negative). In total, 236 rows are removed from the final table.

The original ids in ERD tables *listing_info* and *host* are safe to be dropped now. This is also the last phase of implementing database. We can now start with section V – querying and reporting.

V. Querying and reporting

As mentioned in section I of the report, the goal of the project is to support Paris municipality in understanding the (listed) asking average price of short-stay Airbnb listings in Paris, and how selected characteristics of accommodation and host - individually affected this price in past year. From here, the municipality can work out formulated or fixed price ceilings for accommodation from September 2023, which has abundant time to be adjusted in time for Olympics and big events of 2024 in Paris – thereby protecting the rights of tourists and genuine home renters.

With that goal, we explore the price_per_person variable in the dataset step-by-step. The first question was:

5.1. What was the average asking price (per person, per night) of active short-stay listings in Paris?

This question only required the average of *price_per_person* – of which the query to implement the database in section IV already utilized. For that reason, part of the code from section IV is used to calculate price here. On average, each person can expect to pay 63.12€ per night to stay in a short-stay Airbnb accommodation.

The average price above is generic, assuming that all accommodations with same characteristics offer same price and price of each accommodation is stable across days. We already know that this is not realistic, and spend the following questions to uncover the price average when these factors are accounted in. First, based on variance in price alone, we can suspect that price is different on normal days – weekends – holidays due to travel pattern and increased demand for stay. The second question explore this variance:

5.2. What was the average price of Paris listings, during weekends/holidays/normal days?

- **Consequently, what is the expected markup percentage for special days for price?**

We have to categorize price of a listing on a given day to a day type, then calculate average price per person across same type of days. Each listing will have three prices – which we use to calculate average price in Paris in respectively normal days, weekends and in national holidays.

Answering this question requires data from three tables: *lising_info's* (listing id and accommodates), all of data in *day_price* and *special_day*. Because of that, I first created an index for *listing_info* with two relevant variables, and then LEFT JOIN three tables in the above order, on matching foreign key. The result is 32664 listings with accommodation info * [1;365] row of info about price, including the type of day for price. From this temporary table I calculated AVG price of all listings grouped by *listing_id* and *day type*, so the resulting query is each listing's price (AS *listing_price*) over three types of day. The resulting query is used as a temporary relation named *temp_price_by_day*. Now that *temp_price_by_day* has full info about 32664 listings' price over three

types of day, I only need to take the AVG of *listing_price* from the table GROUP BY *special_day*. Table 1 presents the average price on weekdays/weekends/holidays of Paris listings, rounded to two decimals:

Table 1

Average price per person for Paris listings – based on day type (overall)

Type of day	Price
Weekdays (normal)	63.01
Weekends	63.28
Holidays	63.62

The price does not substantially different based on weekends and holidays, across all Paris listings. They are within the range of average price, with at most 60-cent difference (per person, per night). While we can argue that the difference is multiplicative with group size and number of nights staying, the difference is not very noticeable and cannot be easily felt.

However, we have yet to consider the heterogeneity of markup price behavior. Many hosts set a fixed price for their accommodation throughout the year, but there are also those who flexibly markup the price based on specific day to cater for tourists/travelers. To filter out only those who practice markup behavior requires some adjustment in query code: The same *temp_price_by_day* is first created. It is then LEFT JOIN with exact replica by matching id, resulting in a table in this format:

temp.id	temp.price	temp.day_type_	t.id	t.price	t.day_type
1	X	Normal	1	X	Normal
1	X	Normal	1	Y	Weekend
1	X	Normal	1	Z	Holiday
1	Y	Weekend	1	X	Normal
1	Y	Weekend	1	Y	Weekend
…					
2	X	Normal	2	X	Normal
…					

(Note: In DB Browser shows no prefix before variables – I assume application feature. temp_id and t_id should also have been as one column instead of two in DB Browser)

Which allows me to compare price of different day types for each listing. For a listing (when temp_price_by_day.id matches t.id), if $X \neq Y$ or $Y \neq Z$ or $Z \neq X$, the id, price and day type of that row is selected in return query table. Each average price of a property's day type is compared to 3 pairs of price on day type (including itself), therefore an unequal expression will always appear twice, and needs to be de-duplicated. The result query is used as second temporary relation named

markup_listing, containing all listings with markup price behavior, from which I calculate AVG price GROUP BY special day. Table 2 presents updated average price per person per night for Paris listings that has markup price behavior:

Table 2

Average price per person for Paris listings – based on day type (only for listings doing markup price)

Type of day	Price (in euro)
Weekdays (normal)	68.43
Weekends	68.86
Holidays	69.39

The difference is now a little more noticeable, with 1 euro per person per night more in holidays compared to weekdays. This is still, however, equivalent to at most 1.46% (0.96 €) increase in base price – which can arguably be negligible for short stays. What is worth to consider, though, is that listings with markup behavior have in general higher average price than stable listings. With given information from above queries, we can work out that stable listings ask for, on average, around 53.5€ per person per night. Compared to holiday price of markup listings, this is a 29.7% decrease, which is noticeable. Therefore, we can determine that markup percentage do not change the price of listings drastically to be a factor in the price ceiling; but the fact of whether the listing has markup behavior or not is.

Since markup price is not noticable (although baseline price difference between stable listing and markup listing exists), we can disregard that price fluctuates based on type of day. This is convenient, since we can use calculated *price_per_person* from the first question to answer the upcoming questions. Starting with a pair of similar questions:

5.3. What is the difference in average price of Paris listings, based on neighborhood the listing is in? (Query explanation also applies to 5.4)

This question (and question 5.4) requires data from only *listing_info* table: *redacted_listing_id*, *price_per_person*, *neighbourhood_cleansed*, *room_type*. Therefore, I created another index on the table with the mentioned variables.

Both queries is simply ROUND AVG of prices (to two decimals), GROUP BY listing characteristic variable (*neighbourhood_cleansed* for Q3, and *room_type* for Q4). A count for characteristic variables is also utilized to detect any imbalance.

Table 3 presents the result of average price by neighbourhood, accompanied by graphics & count distribution:

Table 3

Average price per person for Paris listings – based on neighbourhood

Neighbourhood	Price (€)	Count
Ménilmontant	38.94	1706
Buttes-Chaumont	39.07	1791
Reuilly	41.76	2132
Gobelins	45.98	1293
Buttes-Montmartre	51.04	2872
Observatoire	52.86	1397
Popincourt	54.8	2172
Batignolles-Monceau	58.31	2463
Entrepôt	60.63	1747
Vaugirard	62.73	2229

Neighbourhood	Price (€)	Count
Passy	63.22	2813
Temple	74.14	1454
Bourse	76.91	1257
Panthéon	78.06	1003
Hôtel-de-Ville	82.19	1072
Opéra	86.19	1424
Palais-Bourbon	91.23	784
Luxembourg	94.1	937
Louvre	104.57	837
Élysée	109.43	1045

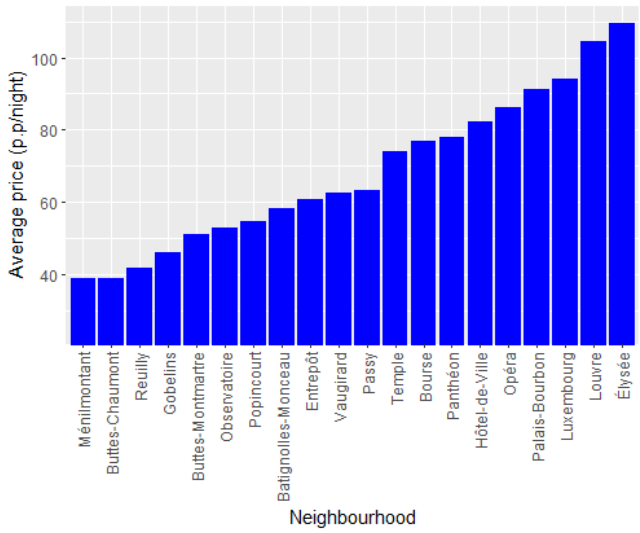

Figure 1: Average price per neighborhood

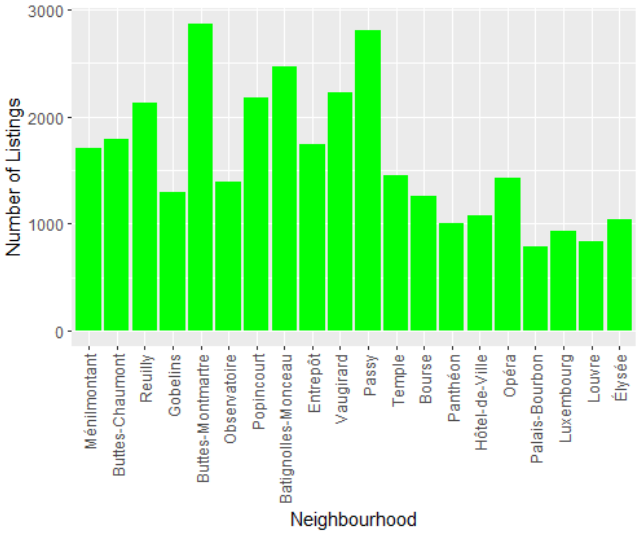

Figure 2: Count of listings per neighborhood

From the table and figures, the following observations can be made:

- Price can be divided in three brackets: 9 central districts of Paris (with highest price), 4 residential/villagery districts (with lowest price), and the 7 middle in-between districts.
- Price difference between brackets are noticeable: The highest price in lower bracker and lowest price in higher bracket is still ~10% (6€ - 10€) jump in average price per person, per night.
- Price somewhat correlates with demand, based on graphs. Central districts with highest price also has fewest active listings; in-betweens have a lot more listings. Residential/ villagery districts have moderate amount of listings.

Coming back to price ceiling, Paris municipality needs to consider based on the brackets the listing is in to set realistic expectations. Also a caution: For the upcoming Olympic 2024 the number

of active listings will increase drastically across districts to meet demands – except for the central area that is already utilized for social-cultural purposes. With increased demands and same supply of accommodation, expect a sharp increase in price in central Paris, and adjust ceiling accordingly.

5.4. What is the difference in average price of Paris listings, based on room type of the listing?

Table 4 presents average price per person, per night of Paris listings, based on room type:

Table 4

Average price per person for Paris listings – based on room type

Room type	Price (€)	Count
Shared room	46.2	201
Entire home/apt	57.85	26971
Private room	78.4	4464
Hotel room	160.61	792

Result is expected, when taking into account the data context. Shared room and hotel room are cheapest and most expensive, respectively. It might seem strange that each person staying in entire home/apartment pays less than in a less-convenient private room, but if the above query included average accommodation size of listing (there is one in the code) you would see that private room has smaller size of accommodation than entire homes. When understanding that entire home/apartment will host more people on average than private rooms, the smaller price in the former is simply due to dividing to a bigger accommodation size.

As we finish exploring average price per person, per night based on accommodation characteristics, we turn to price based on host characteristics:

5.5. What is the difference in average price of Paris listings between superhosts and normal hosts?

This question is similar to previous questions on listing characteristic. The following data are needed: *listing_info*'s *redacted_host_id*, *price_per_person* and host's *host_is_superhost*. The index *listing_host* is first created to contain relevant variables from *listing_info* table. When LEFT JOIN the listing_info table with host table on matching id, each row has the average price of listing and whether or not its host_is_superhost. The calculated value is the ROUND AVG price that is GROUP BY host_is_superhost.

Listings from those that are not superhosts ask for 62.55€ per person per night, while listings from those who are ask for 65.52€ per person per night. There is only a 4.7% difference between the two prices, which the Paris municipality might or not take into account in the price ceiling. I would say this is not a large difference and can be ignored anyways.

5.6. What is the difference in average price of Paris listings between professional hosts (having more than one listing) and casual hosts?

For this question, I also need to categorize hosts based on *host_total_listings_count*. This is achieved using CASE when the variable = 1 (host only has one listing total – Casual Host) and else (Professional Host) – which will sort all rows into a temporary column I named *host_classification.* What's left is to retrieve the average price per person, per night of all listings, then calculated the ROUND AVG that is GROUP by *host_classification*. Table 5 presents the result of query:

Table 5

Average price per person for Paris listings – based on host experience

Host classification	Price (€)
Casual host	51.02
Professional host	70.59

Professional hosts ask, on average, 38.4% (29.57€) more than casual hosts for each person, per night. This is indeed a factor to consider when implementing price ceiling.

This is also the last host characteristic (and last characteristic overall) that can cause difference in average asking price. Finally, we consider whether reviews that are quality and/or popular set a higher price than other properties. They are favorably used by travelers and have the popularity/reputation to justify setting a higher price than average; at the same time, price might have been their unique selling point.

5.7. For top 5% of listings in terms of quality (high review score from many reviews), what is the difference in average price, compared to the rest of listings?

All needed variables are in the *listing_info* table: *redacted_listing_id*, *review_scores_rating*, *number_of_review*, *price_per_person*. Once again, creating an index *listing_review* to shorten query time (and reduce complexity).

Before getting into the actual query, we need to define what constitutes top 5% of listings - defined from all active listings (32428), and is equal to 1621. A high review score signals quality of a listing, but a 5.0 from 1-5 reviews does not signal consistency. It might be that the listing hasn't received its rating on a bad day yet – and we need a threshold to make sure that the high quality is also consistent across different types of stayers. Because of that, I query the average number of reviews from listings that received review before, and found that the number is ~40. Half of the average – 20 – is used as the threshold for a listing to be in quality ranking. Statistically, 20 reviews is also a good threshold, since a property meeting this threshold at the minimum cannot have one mediocre rating of 3, or have many more reviews to balance it out (if they were indeed in the top 5%).

With criteria being established, I first rank all properties based on their review score, then number of reviews (as tiebreaker) using DISTINCT RANK(). Even then, the threshold between top 5% and below still has properties with same rating and same number of reviews – so I elect to put forward the theory of quality listing = higher price and tiebreak using price. All listings in ranking need to have a review score and 20 or more reviews (WHERE clause). Using LIMIT 1621 I can get the top 5% listing, which is put in a temporary relation named *top_listing*.

I can then create a second relation, *non_top_listing*, that is just those with redacted_listing_id NOT IN top_listing. From the two temporary relation for two types of listings, I can calculate average price in each, and UNION them to display result, presented in table 6 below:

Table 6

Average price per person for Paris listings – based on rating:

Category (rating)	Price (in euro)
Top 5%	63.83
The rest	63.08

The price difference between top 5% listings in Paris and the rest is just 1.1% (0.75€). Being popular does not make listings in Paris have a higher price than others. We can, of course, examine the difference of top 5% in sub-groups (neighbourhood, room type, host experience) on price, but the current knowledge at this point provided enough exploration data for the municipality of Paris to build and implement the price ceiling.

5.8. Summary

To summarize, here are the findings of this project, regarding the average price of Paris Airbnb listings:

- Each person can expect to pay 63.12€ per night in a listing. There are factors affecting this price, but they can expect to pay around this range.
- The following factors and characteristics makes a noticeable difference on price:
    - Markup behavior: Listings that change their price on weekends and holiday charges ~30% more than listings with a stable price (53.5€)
    - Neighbourhood (9 central districts: ~75-110€ per night; 4 villagery/residential districts ~39-45€; 7 remaining districts ~50-65€)
    - Room type (Shared room ~46€, Entire home ~57.85€, Private room ~78.4€, Hotel ~160.6€)
    - Host experience: Casual host (~51€) vs Professional hosts (~71€)
- The following factors and characteristics do not make a noticeable difference on price:
    - The actual markup price in weekends/holidays: At most, a 1.46% increase compared to average normal day price.

- Superhost status of Airbnb: 4.7% increase (~65€) for those who are, compared to those who are not (~62€)
- Being quality, as in having average rating over a high amount of reviews: 1.1% increase in average price for top 5%, compared to those who do not (~63€).

With this knowledge, the municipality of Paris can perform some economic analysis to implement price ceiling. If the price ceiling is a function, it should be expressed as:

$$P = fP(N, R, H, A, M, u_P)$$

Where P denotes price, N denotes neighborhood bracket, R denotes room type, H denotes categorization of host into experience and casual, A denotes accommodation size, M denotes whether the listing's price fluctuates or stables in the year, and u_p denotes disturbance factors (such as inflation, license, specific area, host anomaly behavior, and more).

If the municipality of Paris decides to use fixed price ceiling instead, neighbourhood, room type and host experience should be more suitable criteria than just listing size. If use one of the above-mentioned as criteria, adjust the expected average per person to account for fluctuations between a year, and use as reference when checking the price, accommodation size, and criteria of each property. If using more than one criteria for fixed price ceiling, or if prefer to extend the exploration to beyond Airbnb and to accommodation platforms in Paris, please consider to expand the scope of the project.

Finally, a word of caution: All of the figures obtained from this project should only be used as a reference to the structure of the price ceiling, and not be taken literally. The prices reflected are from September 2022 – September 2023, which is already a bygone period. The price are locked in, static at the calendar scrape day – many events, and development has happened in Paris to shake up the price. However, in the volatility of demand, supply and price for upcoming Olympic 2024, an understanding of stable price structure will support the Paris municipality in creating a fair, efficient price ceiling that protect both the rights of tourists and homeowners.

Reference list

Airbnb. (2023, August 1). *Unprecedented interest in the Paris Region during Olympic Games Paris 2024* [Press release]. https://www.hospitalitynet.org/news/4117537.html

Airbnb. (n.d.). *How to become a Superhost - Airbnb Help Centre*. https://www.airbnb.com/help/article/829/

CESSDA Training Team (2020). *CESSDA Data Management Expert Guide*. Bergen, Norway: CESSDA ERIC. DOI: 10.5281/zenodo.3820473

Gadeva, E. (2023, July 26). Parisians expect to cash in with Airbnb during 2024 Olympics despite tighter regulations. *France 24*. https://www.france24.com/en/france/20230726-parisians-expect-to-cash-in-with-airbnb-during-2024-olympics-despite-tighter-regulations

Gibbs, C. H., Guttentag, D., Gretzel, U., Morton, J., & Goodwill, A. M. (2017). Pricing in the sharing economy: a hedonic pricing model applied to Airbnb listings. *Journal of Travel & Tourism Marketing*, *35*(1), 46–56. https://doi.org/10.1080/10548408.2017.1308292

InsideAirbnb. (2023b). *Explore | Paris*. Retrieved September 27, 2023, from http://insideairbnb.com/paris

Inside Airbnb Data Dictionary. (n.d.). Google Docs. Retrieved September 15, 2023, from https://docs.google.com/spreadsheets/d/1iWCNJcSutYqpULSQHlNyGInUvHg2BoUGoNRIGa6Szc4/edit#gid=1322284596

IT Governance Privacy Team (2020): *EU General Data Protection Regulation (GDPR) – An implementation and compliance guide, fourth edition.* IT Governance Ltd.

Lloyd, O. (2023, August 3). *Airbnb to notify users of unfair pricing during Paris 2024 after Government bill*. Retrieved September 11, 2023, from

https://www.insidethegames.biz/articles/1139514/france-government-airbnb-paris-2024

Moreno-Izquierdo, L., Egorova, G., Peretó Rovira, A. and Más Ferrando, A. (2018), "Exploring the use of artificial intelligence in price maximisation in the tourism sector: its application in the case of airbnb in the valencian community", Journal of Regional Research, Vol. 42, pp. 113-128.

PublicHolidays.fr. (2022). *France Public Holidays 2022*. Retrieved September 11, 2023, from https://publicholidays.fr/2022-dates/

PublicHolidays.fr. (2023). *France Public Holidays 2023*. Retrieved September 11, 2023, from https://publicholidays.fr/2023-dates/

Priante, A. (2023). *Cleaning with SQL.* Retrieved September 26, 2023, from https://canvas.eur.nl/

Similarweb. (n.d.). *Top Travel & Local Apps ranking - Most popular travel & local apps in France*. Retrieved September 11, 2023, from https://www.similarweb.com/apps/top/google/store-rank/fr/travel-local/top-free/

Qiu, R. T., Fan, D. X., & Liu, A. (2017). Exploring the booking determinants of the Airbnb properties: an example of the listings of London. In *Springer eBooks* (pp. 44–51). https://doi.org/10.1007/978-3-319-72923-7_4