Data Management and Ethics

Individual Integrative assignment

Data Management for good: the social and economic plan of Airbnb

Remy Dinh – 525000rd

10-8-2023

Contents

Task 1 Plan:

Rome, the capital of Italy, is renowned for its rich historical heritage. It's great history, cuisine, architecture, art makes it a top travel destination for many tourists in the world. Despite the fact that the tourist industry brings in a lot of value to the city and its citizens. Mass tourism also has it's flipside. One ongoing concern is the constantly growing amount of Airbnb's. Neighbourhoods with many Airbnb's are experiencing tourist disturbance (noise, drunk people at night), see more and more restaurants/shops appear that are purely catered towards tourists and finally, the increasing amount of vacation rentals is also causing prizes real estate prizes to grow quickly in these locations. These factors makes many neighbourhood's in Rome unliveable for actual Romans and has caused many to leave to city. This entire phenomenon is described as the desertification of Rome (Reguly, 2023). To address this issue the municipality of Rome has banned the listing of new Airbnb's from some of their historic neighbourhood's already (Symons, 2023), but you could argue that this might lead to a shift in locations of these rentals rather than a solution for desertification in Rome as a whole. That is why this report will explore another solution to Airbnb desertification already implemented by many other cities (e.g Paris, Amsterdam (Symons, 2023) being: A limit of the amount of nights a property can be used for Airbnb renting. This solution makes the business case of running Airbnb's less attractive, limits the amounts of tourists that can stay in certain neighbourhood's and thereby addresses desertification. To explore whether this solution is viable for the city of Rome and to determine what the limit should be the following research question is defined:

Could a limit in allowed rental days for Airbnb's be a solution for the degradation of certain neighbourhoods in Rome and if so in what form?

To work towards the answering the aforementioned main research question, sub-questions will be answered with the Inside Airbnb data at hand:

- **How many days a year is a listing rented out in Rome (on average)?**
  - ➔ Relevance: This allows us to see whether Airbnb's in Rome are indeed as popular as assumed and thereby a big cause of desertification. And to determine what the limit should be to be effective to combat the issue.
- **When are Airbnb's in Rome most occupied? (amount of reviews per month)**
  - ➔ Relevance: Exploring the reviews per month allows for the exploration of seasonality amongst Airbnb rentals. As amount of reviews is a good indicator for occupancy. If the properties are mainly rented out during certain periods in the (e.g. holidays/summer), rental might be less effective in combating desertification as nuisance would then be concentrated in certain period. If that is the case a limit per month/for specific period could be explored. Also, looking at the trend for a longer period of time allows to see how the popularity of Airbnb's in Rome has progressed over time.
- **How many Airbnb listings does each identified neighbourhood in Rome have and what is their average occupancy?**
  - ➔ This aggregation allows us to get a better idea of what neighbourhoods are experiencing most disturbance and would benefit most from the implementing the limit.

Task 2 Design & Organize:

Now that there is an understanding of the data and the questions at hand it is of utmost importance explore data structures, relationships and to create a blueprint for the data in the form of an ERD. Each step to get to the creation of the physical ERD and accompanied considerations and challenges will be explored:

1. Entity Identification:

The main two entities of importance in this database are listings and reviews. A third entity, "occupancies" will later be added to comply to normalization rules:

- Listings is a dataset contains a wide array of data for each unique listing (e.g. title, price, neighbourhood)
  - ➔ Relevance to RQ's: contains the data that allows for calculation of occupancy with data on the amount of reviews. (last twelve months) and the minimum amount of nights a guest can stay at a specific listing. The computations will be further explored in Task 3. Moreover, the table also contains data on the different neighbourhood's and therefore allows us to calculate the most popular neighbourhood's and their occupancy.
- Reviews is a dataset that contains data for each unique review made (e.g. date, reviewer id, comment)
  - ➔ Relevance to RQ's: contains data on the date a review is made and for what specific listing the review is for. This will allow us to calculate the amount of reviews per month.
- It will contain a listing identifications number and its accompanied occupancy for the last 12 months.
  - ➔ Insights in the occupancy for each of the listings allows us to calculate the average occupancy in Rome and per neighbourhood in Rome.

2. Relationships and cardinalities:

These connections are apparent between our defined entities:

Listings to reviews (One to many): One listing can have zero, one or multiple reviews, one review always refers to only one listing.

Occupancies to listings (One to one): Each listing can only have one occupancy rate and one occupancy only refers to one listing.

3. Select relevant variables/attributes

In line with the minimization practice of the GDPR (EDPS, 2023) only variables/attributes that were of importance to the answering of the research questions were included and altered if necessary/possible. All included variables/attributes are identified in table 1 and accompanied complex considerations will be briefly introduced in the segment under the table.

Table 1: Summary of relevant variables/attributes

Variable/Attribute	Data type	Entity in which included	What is it?	Why included?	Primary (PK)/ Foreign Key (FK)?
neighbourhood_cleansed	text	listings	Neighbourhood determined using coordinates of the listing	Allows for calculations per neighbourhood. (amount of reviews & avg. occupancy)	-
number_of_reviews_ltm	integer	listings	Number of reviews for a listing in the last twelve months	Used in the calculation of occupancy (will be further explained in task).	-
minimum_nights	integer	listings	Minimum amount of nights you have to for book at a listing	Used in the calculation of occupancy (will be further explained in task).	-
id_listing	integer	listings, reviews, occupancy	Unique identifier for each listing	Allows for calculation of the amount of listings per neighbourhood and occupancy per listing. Makes all tables possible.	PK & FK
id_review	integer	reviews	Unique identifier for each review	Allows for calculation of the amount of reviews per time period. Makes joining table listings and reviews possible.	PK & FK
month (from date)	integer	reviews	Month review was placed	Allows (together with year) for calculation of reviews per time period (month – year)	-
year(from date)	integer	reviews	Year review was placed	Allows (together with month) for calculation of reviews per time period (month – year)	-
occupancy	integer	occupancy	Amount of days a listing was occupied the last 12 months (calculations in task 3)	Allows for calculation of average occupation and average occupation per neighbourhood in Rome.	-

Complex considerations:

Choice neigbourhood_cleansed:
The dataset had several options for determining the neighbourhood a listing was in. Both viable attributes "neigbourhood" and "neigbourhood _group_cleansed" contained a lot of missing values and attribute "neigbourhood" specifically gave very general identifications for the neigbourhood. The chosen attribute: "neigbourhood_cleansed" had a well specified value for each listing and was therefore decided on.

Choice minimum_nights:
To get a value for the minimum amount of nights a guest has to book a certain accommodation. Two options were presented in the set. "Minimum_nights and "minimum_nights_avg_ntm". The first one presents the minimum nights based on the single value that the owner determines, the latter is based on what is average minimum stay identified in the future calendar table. As the research is based on a period in the past and calculations on an continuously/variable calendar could be prone to inconsistencies the stable "minimum_nights" variable was chosen.

Id_listing & id_review (Primary & Foreign key):
Even though both "id_listing" and "id_review" can be considered personal data as the id's could directly or indirectly identify an individual. They are of utmost importance in the process of identifying individual listings and reviews and performing calculations with them and can therefore not be excluded. Pseudonymization for both id's should be looked at during the cleaning process.

Additionally, "id_listing" and "id_review" are both primary and foreign keys in our model. id_listing is a unique value in both the occupancies and listings table and is used as primary key in both. Id_listing is also included in the reviews table as a foreign key and is the connecting factor between tables listings and reviews. Lasting id_review uniquely identifies each review in the reviews table and will be used as primary key.

Month & Year

Month and year will be extracted from the date variable in the reviews table. As only the period month-year is relevant in the scope of our research, days is excluded. In line with improvement of anonymization of the date.

3. Normalization

The dataset at hand does require some normalization. The normal form (NF) 1-3 will be examined.

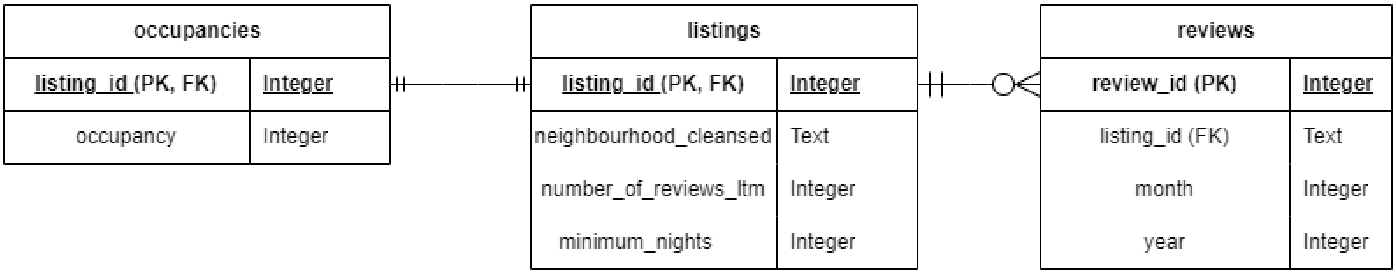
- 1NF: the date variable in the reviews table contains values for day, month and year and is therefore not atomic. Splitting up the components was done to conform to 1NF and simultaneously as it makes analysing individual parts and leaving out the day value easier. Also, both the listings and reviews table has a column name id that does not refer to the same variable. To comply to 1NF these have to be made unique also, this will be done in task 3.
- 2NF: 1NF restrictions are met and the primary keys consist out of individual columns, so 2NF is adhered to.
- 3NF: occupancy depends on "number_of_reviews_ltm" and "minimum_nights". Therefore including it in the same entity would cause redundancy. Every combination of

“number_of_reviews_ltm” and “minimum_nights” has the same occupancy. Therefore the extra column for occupancies was created.

4. ERD:

Putting all of the previous considerations together, results in the following ERD:

Figure 1: ERD



occupancies	
listing_id (PK, FK)	Integer
occupancy	Integer

listings	
listing_id (PK, FK)	Integer
neighbourhood_cleansed	Text
number_of_reviews_ltm	Integer
minimum_nights	Integer

reviews	
review_id (PK)	Integer
listing_id (FK)	Text
month	Integer
year	Integer

Task 3 Data Processing:

Processing:

It was decided to perform all data processing tasks using R due to personal ease of use and experience.

Before starting to check for quality, cleaning the data and accounting for personal data. Some processing of the original data set had to happen related to the "occupancy", "year" and "month variable":

- **Occupancy**:

The definition of occupancy in this research is: the amount of days an Airbnb lisitng was occupied in the last 12 months. This was done using the "San Francisco Model" developed by Inside Airbnb (n.d.). The particular assumptions made for our analysis:

Formula: Occupancy = Review rate * number of reviews * average length of stay

- Review rate: conversion rate of stays to reviews. This was assumed to be equal to 50%. Based on comparisons of research done by Inside Airbnb (n.d.). Meaning that for every review 2 stays are expected. Review rate = 2
- Average length of stay: This was assumed to be 3.3 nights . Airbnb (2016) published an average stay length in Italy of 3.6 nights another report by Buzzacchi et al., (2020) reported a value of 3.48 for this for 2018. The national institution gave an average length of stay at hotels/comparable accommodations in Italy of 2,93 (Istat, 2018). The mean of these three values was taken as an estimate. If minimum nights was more than 3.3, the minimum nights values was taken on as the average length of stay. To allow for more accurate estimates.
- Occupancy cap: To account for high outliers in minimum nights (long stay) and number of reviews (popular listings) and to keep the model conservative a cap of 70% (255 days) was used, equal to a occupancy for a popular hotel in line with Inside Airbnb assumptions (Inside Airbnb, n.d.).

- **Year and month (from date)**

To comply to 1NF the "date" variable had to be split up as decided on in the normalization segment of question 2. Moreover, as the scope of the analysis only requires for calculation on month and year, the day value was not included anymore in line with the anonymization principle.

Quality check:

Explore tables:

First the tables with the data were visualized and looked into. It was seen that at this stage the column name "id" in the listings and reviews table referred to something different. "id" in listing referred to the id of a listing whereas "id" in review referred to an "id" of a review. To create more clarity and prevent mistakes in computations. The unique id's for listings and reviews were renamed listing_id and review_id.

Summary tables:

Thereafter summary tables were created for the 3 datasets at hand (listings, reviews, occupancy) to explore whether inconsistencies exist in the data of interest that should be accounted for. Some important insights:

- None of the tables had missing any missing or NA values.
- None of tables had (unexpected) negative values.
- The listings table, number of reviews and minimum nights have very few large outliers with max: 621, max: 999 respectively. The aforementioned variables are solely used for calculating and their outliers are accounted for in the creation of the occupancy variable by capping occupancy at 70% (as explained in the previous part). Even though that this approach might be rather simplistic and might still lead to some inaccurate values for individual cases, as the aim of this research is to perform analysis only aggregates of the data, leading to ignorable impact of these individual values. Also, deleting or adjusting records with high number of reviews or with a large minimum nights of stay would make the research less representable of reality, as popular listings and long-term Airbnb's would not be taken into account/be accurately represented anymore whilst they are relevant for the scope of our research.

Look whether the id's used as PK were unique:

The "unique" function in R was used to determine whether there were duplicates among the primary key (PK) columns in the dataset. For the listing table no duplicates for listing_id were found, the selected primary key is unique. In the review table, one duplicate was apparent for review_id = 764310348900092416. As it only relates to two rows, it was decided to fix this inconsistency by deleting both records.

Pseudonymization

As for the analysis personal data is used in the form of listing_id and review_id, pseudonymization is required. To ensure uniqueness the pseudomized id's and allow for simple implementation, a sequential id was assigned for all id's in lasting and review_id. (e.g first id in the table= 1, second id " = 2, third id " = 3 etc.). As the order the order does not give away information for neither listing_id or review_id, this technique is a good fit in our case. A pseudonymization mapping table for both of the listing_id and review_id was created. The original listing_id's/review_id's were replaced with the pseudomized ones in the listings, reviews and occupancy table.

Task 4 Database implementation:

After the cleaning process performed R is, the table structure should be created in SQL and populated using the cleaned tables created in task 3. The syntax can be found under the “execute SQL tab” under the name “implement_populate”.

Implementation:

For each variable/attribute in each of the tables the data type and constraints were locked in within the CREATE TABLE function. The data types was “INTEGER” for all variables and “TEXT” for only “neighbourhood_cleansed”. Moreover, all variables were given the “NOT NULL” constraint as we do not accept any empty cells in our tables as all selected variables/attributes are required for computations. The “NOT NULL” constraint did not have to be applied to the primary keys as primary keys inherently cannot be empty.

Listings is the parent table in our database, all foreign keys in our design refer back to the primary key “listing_id” in this table. “Listing_id” is the primary key of this table and was given the constraint PRIMARY KEY, that automatically checks for uniqueness and non-null values.

Occupancies is one of the child tables in our database, a foreign key constraint applies. To be able to identify the one to one relationship to the listings table. The primary key, foreign key and reference constraints were all set to listings_id in the CREATE TABLE argument.

Reviews is one of the child tables in our database, a foreign key constraint applies. To be able to identify the many to one relationship to the listings table. The primary key was set to “review_id” and the foreign key and reference constraints were both set to “listings_id”in the CREATE TABLE argument.

Populating:

After implementation was completed, all tables in csv format created in task 3 were loaded into DB browser including the pseudo mapping tables for listing_id and review_id. The listings, reviews and occupancies table were populated with the specific attributes/variables defined the physical ERD. In this process no errors were observed, meaning that the imported data conformed to the data type and constraint specified for each variable/attribute. No further actions will be performed on the pseudo mapping tables (pseudopmap_listingid and pseudomap_reviewid) but they will be kept within the database environment for completeness and for possible future de-pseudonymization.

Task 5 Querying and Reporting:

1. QUESTIONS

In this section the questions defined in Task 1 and displayed once again on this page, will be addressed using queries and their results.

Main question: ***Could a limit in allowed rental days for Airbnb's be a solution for the degradation of certain neighbourhoods in Rome and if so in what form?***

Sub questions:

1. **How many days a year is a listing rented out in Rome (on average)?**
2. **When are Airbnb's in Rome most occupied? (amount of reviews per month)**
3. **How many Airbnb listings does each identified neighbourhood in Rome have and what is their average occupancy?**

Explanation of the Queries

The syntax for all queries substantialized in this segment can be found under the second tab called "Queries" under the 'Execute SQL' tab or in the Task 5 section of the .txt file.

To provide an answer to sub-question 1, the view "average_occupancy_Rome" was created because this allows for easier visibility and reusability of the result. The view contained the AVG function on the variable "occupancy" from the occupancies table to calculate the average occupancy across all listings in Rome as asked for in sub-question 1. The AVG function was chosen as this is an well-known integrated function within SQlite that allows for direct calculation of averages. Other methods (e.g. manual calculations or combination sum/count) would make the query unnecessarily complex and deliver the same outcome.

To provide an answer to sub-question 2, the view "year_monthly_reviews" was created. The view contained a SELECT clause with two elements taking information from of the reviews table. The first element makes use of the concatenation function "||" to combine values for Year and Month to one element with a "-" in between. The "printdf" function is used the nicely format Month so that it always consists out of 2 numbers (e.g. 5 becomes 05). This formatted combined value of Year and Month is defined as "year_month". The second element is a counter named "review_count," created using the function COUNT. Its purpose is to tally the number of rows corresponding to each of the previously defined year-month combinations. But it cannot do this without the help of the GROUP BY clause that demands that every count calculation is performed for each unique value for year_month". No ORDER BY clause was necessary as the values were already put in chronological order. The identified query addresses sub question 2 as it aggregates the amount of reviews per month for several years. By combining this with the theory that suggests the number of reviews is a reliable indicator of occupancy (task 1), it becomes possible to analyze and visualize the seasonality of Airbnb occupancy in Rome and make inferences about it. Instead of using concatenation for the year-month expression, the DATE function could also have been used. But was decided against because (1) our year and month variables were not in date format but in different columns, (2) it makes the code more complex and storage heavy.

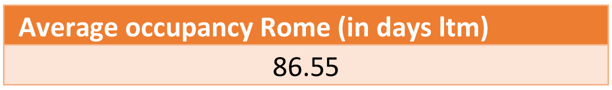
To provide an answer to sub-question 3, the view "occ_nrlistings_neighbourhood" was created. The view selects the neighbourhoods from the listings table, average occupancy from the occupancies table and counts of the amount of rows from the listings table. As information is gathered from multiple tables, an INNER JOIN is necessary between the occupancy and listings table using the primary key/foreign key listing_id. Thereafter the GROUP BY function was used to perform the calculations within the select function, for each neighbourhood. Lastly, the ORDER BY function was used to have the rows organised in a descending order in terms of amount of listings. The identified query addresses sub-question 3 as it puts out the average occupancy for each neighbourhood, as well as the amount of listings for each neighbourhood. An INNER JOIN was chosen as we only want to combine rows that are about the same listing and therefore have the same listing ID. As we know from the data processing step that each listing has a related occupancy and visa vera LEFT JOIN or CROSS JOIN would also have delivered the same results. However, using these techniques could lead to inaccuracies when the dataset gets altered and not only exact matches are apparent, INNER JOIN is also considered a more efficient operator as it only performs calculations on exact matches.

Results with visuals

The results for each sub-question will be explained and altogether reflected on in the conclusion to give an answer to the main question. Visualizations (tables and charts) were created using excel.

SQ1: **How many days a year is a listing rented out in Rome (on average)?**

Table 2: Average occupancy table



Average occupancy Rome (in days ltm)
86.55

According to my estimation an average Airbnb in Rome is rented out nearly 87 days a year. So percentage wise that is about an 86.55/365 = 23.7% average occupancy rate. This number is not that high compared to other cities (SOURCE). However still much higher than the renting out limit of in for instance Amsterdam (30 days). Keep in mind that we are working with averages here, which can be very misleading. It could for example be that there are many unsuccessful listings with 0 reviews that are bringing down the average occupancy or it could be that occupancy is highly neighbourhood dependent. The latter will be explored in SQ 2.

SQ2: **When are Airbnb's in Rome most occupied? (amount of reviews per month)**

Two graphs were created with the query result from subs-question 2. Figure 1 displays the amount of reviews that were published each month from May 2021 until May 2023. The amount of reviews is used as a measure of occupancy of Airbnb's in that month. Figure 1 does show that seasonality plays a role. **The winter period is significantly less popular in terms of occupancy (around -50%), whereas spring, summer and autumn perform relatively stable.** Occupancy is not really concentrated in a small period of the year and a yearly occupation limit could therefore can still be effective in combating desertification, limits per month/specific period are therefore deemed unnecessary.

Also a general upward trend is seen of Airbnb's popularity in figure 1, possibly due to recovery from the COVID 2019 crisis that had a significant impact on the tourist industry as whole (SOURCE). Figure 2 also confirms this with a steep drop in the 2020-2021 period and shows overall trend of Airbnb's growing popularity (in terms of reviews) over the period 2010-2023. The fastly growing popularity of Airbnb in Rome should be a warning for the city as further desertification could become even a larger problem, with more incoming tourists via Airbnb.

Figure 3:

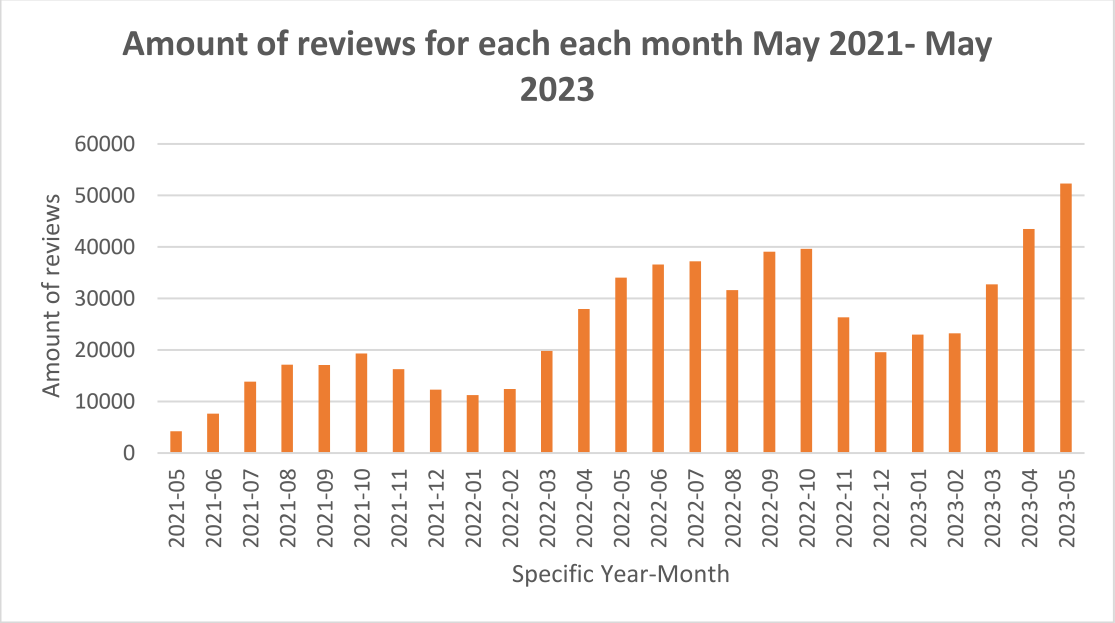
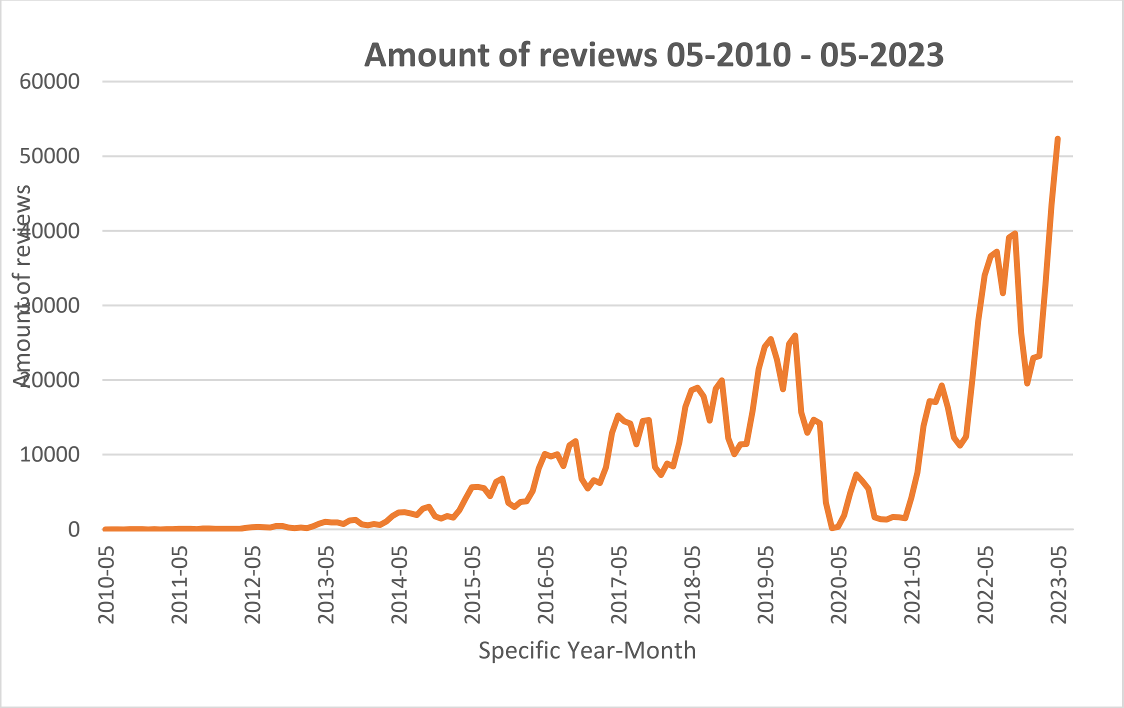


Figure 4:



SQ3: **How many Airbnb listings does each identified neighbourhood in Rome have and what is their average occupancy?**

Table 3: Neighbourhood table

Neighbourhood	average occupancy	Amount of listings
I Centro Storico	*104.11*	14370
VII San Giovanni/Cinecitta	*67.30*	1857
II Parioli/Nomentano	*65.64*	1833
XIII Aurelia	*83.81*	1583
XII Monte Verde	*70.61*	1308
V Prenestino/Centocelle	*82.06*	1027
VIII Appia Antica	*64.31*	746
X Ostia/Acilia	50.71	708
XIV Monte Mario	57.07	520
XI Arvalia/Portuense	52.49	465
XV Cassia/Flaminia	41.93	446
IV Tiburtina	52.48	445
III Monte Sacro	45.78	407
IX Eur	44.44	331
VI Roma delle Torri	39.99	210

Table 3 shows the average occupancy for each identified neighbourhood In Rome, along with the amount of listings apparent in that neighbourhood. The rows are ordered from highest amount of listings to lowest. So this table could also be considered a ranking. The most striking observation that can be drawn from the table is that a vast majority (14370) of the Airbnb's are located in I" Centro Storico" (the center of Rome), this neighbourhood also has a very high average occupancy compared to other neighbourhoods. Additionally, it can generally be seen that the higher amount of listings a neighbourhood has, the average occupancy is.

Taking the answers of the sub-questions with me, I attempt to answer the main research question:

Conclusion:
Could a limit in allowed rental days for Airbnb's be a solution for the degradation of certain neighbourhoods in Rome and if so in what form?

An Airbnb renting limit could definitely be a solution of desertification (as explained in task 1) of neighbourhoods in Rome. As considerable average occupancies could be seen (table 2&3), especially for the city centre and because the popularity of Airbnb is continuously growing and could potentially make the problem worse without further intervention (figure 4). If a rental limit would be imposed, a limit per year would be most sensible as occupancy is not concentrated during small periods in the year but rather stable throughout the year (figure 3). Additionally for the limit to have significant impact for the city centre, it should be at least be lower than the average occupancy in the city centre which is the neighbourhood most listings and the highest occupancy and thereby most nuisance, so the limit should be at least less than 104.

The exact determination of the limit should take into account many other socioeconomic and political factors and can therefore not be determined in the scope of this descriptive research. But overall, it can be concluded that a renting limit could be solution for the problem of desertification of Rome that should be further looked into.

Bibliography:

Airbnb. (2016). Overview of the Airbnb Community in Italy. In *https://www.airbnbcitizen.com/*.

Buzzacchi, L., Governa, F., Lacovone, C., & Milone, F. L. (2020). *The uneven diffusion of shortterm rental markets between urban locations and selective tourism destinations*.

EDPS. (2023, October 2). *Data controller*. European Data Protection Supervisor. https://edps.europa.eu/data-protection/data-protection/glossary/d_en#:~:text=Data%20minimization&text=The%20data%20minimisation%20principle%20is,for%20which%20they%20are%20processed%22.

Inside Airbnb. (n.d.). *Data assumptions*. http://insideairbnb.com/data-assumptions/#:~:text=Inside%20Airbnb's%20%22San%20Francisco%20Model,and%20co%2Dfounder%20Brian%20Chesky.

Istat. (2018, November 27). *Tourist flow in Italy*. https://www.istat.it/en/archivio/224433

Reguly, E. (2023, June 6). Overloaded by Airbnbs and mass tourism, Rome fears its historic centre will be emptied out of locals. *The Globe and Mail*. https://www.theglobeandmail.com/world/article-rome-tourism-airbnbs-locals/#:~:text=The%20defenders%20of%20the%20city,between%2C%20fewer%20and%20fewer%20Romans.

Symons, A. (2023, September 8). Italy, Austria, Malaysia: Which cities and countries are cracking down on Airbnb-style rentals? *Euronews*. https://www.euronews.com/travel/2023/06/11/italy-malaysia-usa-which-cities-and-countries-are-cracking-down-on-airbnb-style-rentals