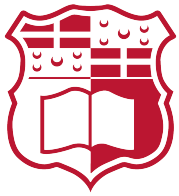

Data Science: From Data to Knowledge

CPS3235 Assignment

QUENTIN FALZON

Supervised by Dr Jean Paul Ebejer

Department of Computer Science
Faculty of ICT
University of Malta

January, 2021

A study-unit assignment submitted in partial fulfilment of the requirements for the degree of B.Sc. Computer Science.

Statement of Originality

I, the undersigned, declare that this is my own work unless where otherwise acknowledged and referenced.

Candidate Quentin Falzon

Signed _____

Date February 20, 2021

Contents

1

COVID-19 Pandemic Analysis

The SARS-CoV-2 virus is a type of coronavirus that causes the COVID-19 disease, characterized by symptoms such as cough, fatigue, fever and breathing difficulties. It was first identified in Wuhan, China during December 2019. Since then, the virus has spread to many countries, including Malta, through human-to-human transmission. The first cases of the disease were officially detected in Malta in the beginning of March 2020. This assignment considers cases from the date of first appearance, to late January 2021.

1.1 Collection, Cleaning and Storage Process

The initial task is to collect data about the COVID-19 pandemic in Malta from multiple sources. Three sources were identified for this assignment, namely:

- **COVID-19 in Malta – Open Dataset**.[1] This is the official open COVID-19 dataset provided by the Public Health Response Team.
- **Covid-19 Malta data**.[2] This dataset is maintained by local company Lobeslab Ltd. using manually gathered data from the daily media update provided by superintendent of public health Profs. Charmaine Gauci.
- **Data on COVID-19 (coronavirus) by Our World in Data**.[3] This compilation of datasets by Johns Hopkins University (JHU) provides global pandemic data sourced from various institutions such as World Health Organisation (WHO) and the European Centre for Disease Prevention and Control (ECDC).

All sources are updated daily with relevant data such as new cases, active cases, recoveries, and deaths, in comma separated value (csv) format. Initially, csv files for each source were loaded into MS Access for a quick overview and comparison of data. Since records are added daily, and COVID-19 has existed locally for just less than a year, it is manageable to manually ensure that figures between datasets are equivalent.

The majority of testing is carried out by the Maltese public health authorities, and all private health facilities are obliged to report test results to them, since COVID-19 is a *notifiable* disease. Consequently, [2] and [3] derive their data from the public health authorities' reported figures [1]. JHU's dataset shows good

[1] https://github.com/COVID19-Malta/COVID19-Cases
[2] https://github.com/Lobeslab-Ltd/covid-19-MT
[3] https://github.com/owid/covid-19-data/tree/master/public/data/jhu

conformity to the data provided by the public health authorities. There are days for which JHU report a slightly different tally of active cases, but the values always converge within 1-2 days.

Some larger inconsistencies were noted when comparing the health authorities' dataset to that of Lobeslab. As of the 16th August, there are discrepancies in the total number of cases. This was seemingly justified by Lobeslab in their dataset repository:

"On Sunday 16th August 2020, the Maltese Authorities announced that cases of COVID-19 from incoming illegal migrants will no longer be included in the total case count. They also reported that past cases will be removed. This has presented a data integrity problem for this repository since it is not entirely clear on which days these cases were removed. We also believe that it is unethical to remove these cases. For this reason, this repository shall not revise the case counts prior to the 16th August. Since we do not have access to the number of cases from migrants, we have no way of including them so as of the 16th August 2020 this repository will not be including migrant cases due to a lack of data from official sources and not by our choice."

Therefore, the significant difference (105) in total cases between the two sources for the 16th of August is justified. Lobeslab stated they will not be including migrant cases beyond the 16th of August, so this difference of 105 was expected to remain constant. However, the difference was observed to increase further on several dates, with no justification as shown in figure 1.1.

Date	Total Cases (Public Health)	Total Cases (Lobeslab)	Difference
2020-08-16	1306	1411	105
2020-08-31	1883	2020	137
2020-09-03	1965	2112	147
2020-09-04	1984	2158	174
2020-09-21	2776	2955	219

Figure 1.1: Inconsistencies between sources [1] and [2] for total reported cases

The difference remains constant at 219 as of the 21st of September. Upon contacting Lobeslab for clarification about the unexplained increasing deltas, it was established that they in fact still have access to data for migrant cases despite claiming the contrary. Therefore, every increase in discrepancy as of the 16th August is due to new migrant cases being announced by the health authorities, but not included in their tallies. Lobeslab have since corrected their dataset description (commit dfa66c2) and fixed some incorrect data entries (commit 32a58cc).

In summary, all the data in [2] includes migrants whereas [1] and [3] stop accounting for them and **remove all past cases** on the 16th August. This sudden omission of data is not representative of the actual situation in Malta (there were 11 recoveries on the 16th, not 105... and definitely not -42 new cases as seen in JHU's dataset). Furthermore, there were occasions (29th – 30th July, 4th – 5th August) where the health authorities excluded migrants from daily cases but re-included them the next day. In view of the inconsistent reporting methods reflected in [1] and subsequently [3], I will be using Lobeslab's dataset to generate descriptive statistics and visualizations about the infection rate in Malta. Therefore, these statistics will

represent a local infected population which is **inclusive** of migrants.

Lobeslab do not include any testing data, so this was taken from [1] and added as a new column to the table. A Linux environment was used to clean and store the collected data. The setup is briefly outlined.

- A MySQL docker container stores tables in a "covid19" relational database.
- *csv* files are imported into Pandas dataframes and parsed according to datatype.
- *SQLAlchemy* is used to bulk import dataframe contents into the MySQL database.

The csv file was imported into a Pandas dataframe. For data validation, dates were parsed as *datetime64[ns]*, country name as a *string*, and all other values as *int64*. Some initial plots were generated to check the data for any noticeable errors. When plotting Total Cases, Total Deaths or Total Recoveries against Date, a persisting outlier was observed, as illustrated in figure X. Clearly, the outlying value is incorrect since cumulative data can spike up but never down. It would make sense for the value to appear further right, with the rest of the data points. Since figures were previously checked to be correct, this error suggested that a correct value had been assigned an incorrect date, with an error of roughly 1 month. Upon examining the data for early November, the problem was found to be a date incorrectly labelled as 2020-11-05 instead of 2020-12-05, resulting in data from 5th December being plotted for 5th November. The erroneously labelled date was manually corrected as part of the reconciliation process. This has also been fixed by Lobeslab (commit dac04c4).

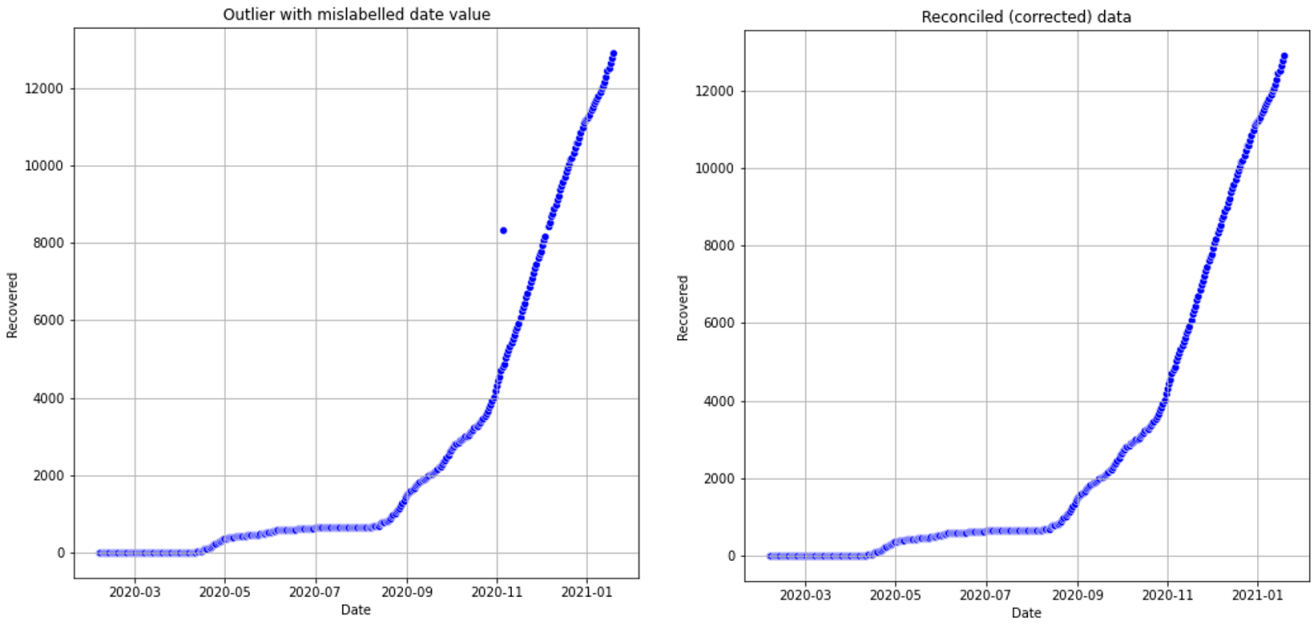

Figure 1.2: Correcting an outlier

1.2 Describing Malta's COVID-19 Pandemic

1.2.1 Infection Rate

The infection rate in Malta can be primarily described by the number of new cases per day (*new_cases*). The number of active cases can also reflect an increasing or decreasing infection rate. In fact, there is a high correlation of 0.92 between new cases per day and active cases. However, the active cases attribute also depends on the number of recoveries, hence it is not solely representative of the infection rate. If there are more daily recoveries than there are new cases, the amount of active cases will decrease, yet it is still possible for the infection rate to rise.

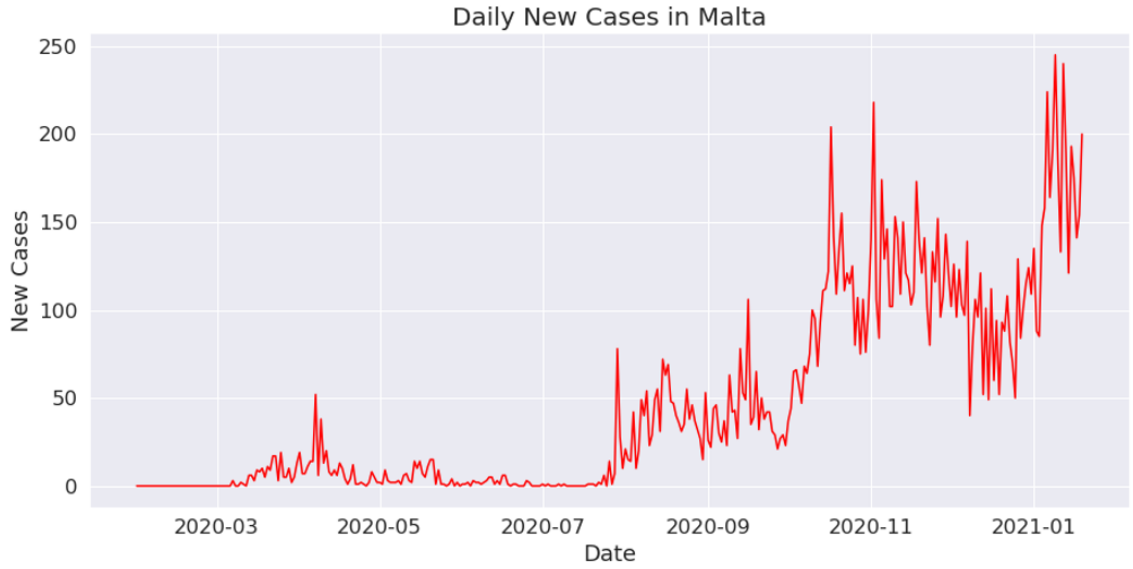

Figure 1.3: Line plot showing number of new cases per day in Malta

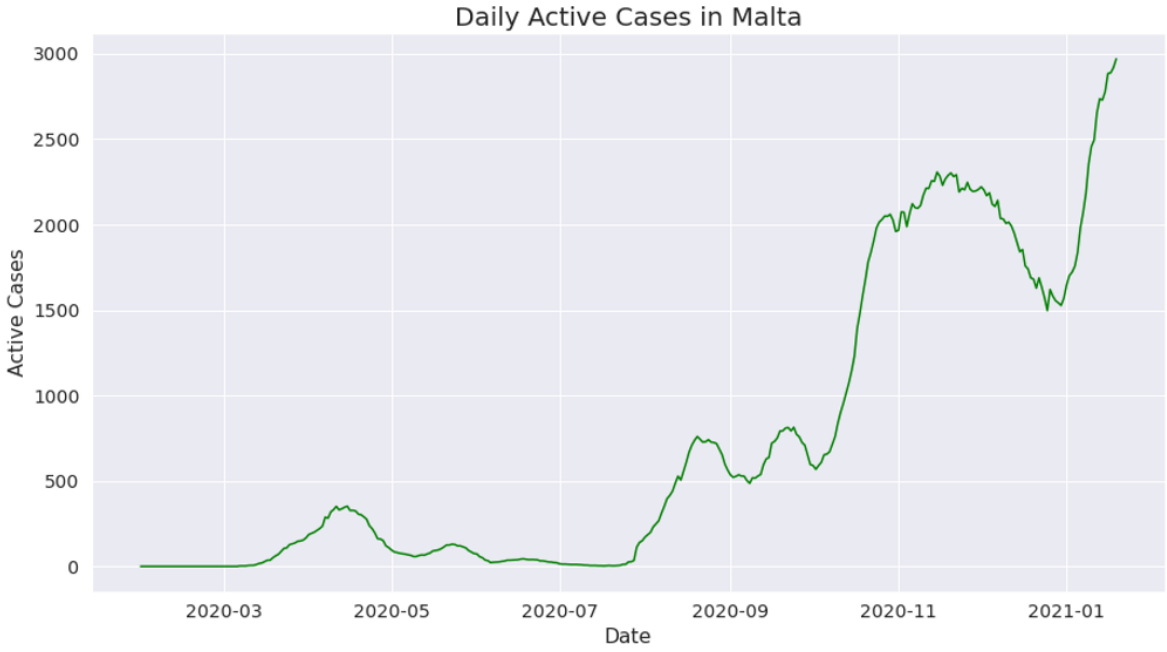

Figure 1.4: Line plot showing number of active cases per day in Malta

Descriptive statistics for the infection rate in Malta and Italy were generated. As indicated by the previous visualisation, Malta's worst day in terms of new infections was towards the beginning of January 2021, with 245 new cases being announced. The minimum value was included for completeness' sake: Both countries had days (when there were already active cases) when there were no new infections. Italy's population is roughly 137 times that of Malta (60.4 million vs 441 thousand). To compare the 2 countries, I normalized each country by population size (obtained from JHU in [3]) and then multiplied by a million to avoid working with very small decimal numbers.

	MT new cases	IT new cases	MT new cases (per million)	IT new cases (per million)
mean	45.41	6800.88	102.85	112.48
median	15	1616	33.97	26.73
min	0	0	0	0
max	245	40902	554.88	676.49
Q1	1	329.5	2.26	5.45
Q3	83	10833.5	187.98	179.18
std dev.	55.63	9857.28	125.99	163.03

Figure 1.5: Descriptive statistics for infection rates in Malta and Italy

I was surprised to see that the means are quite similar between the 2 countries. From what I remember hearing in the news, I was expecting the statistcs for Italy to be significantly worse (Admittedly, I do not follow the news consistently). However, one must remember that when comparing infection statistics bewteen countries, the amount of testing done by each county is assumed to be equivalent. In reality, this is not the case and will be verified shortly. The 1st quartiles show that Malta had more days with relatively fewer new cases per million (2.26 vs 5.45). Conversely, the 3rd quartiles show that Malta had more days with relatively higher new cases per million (188 vs 179). Malta has a lower standard deviation, meaning its cases are more concentrated around the mean than Italy's.

Box plots are great at summarizing large amounts of data. They illustrate how the data is distributed, and show any outliers affecting the measures of location. From the generated box plot it is abundantly clear that Italy has many outliers. Therefore, with the exception of the minimum and maximum values, Italy's statistics shown in figure 1.5 are skewed for the worse.

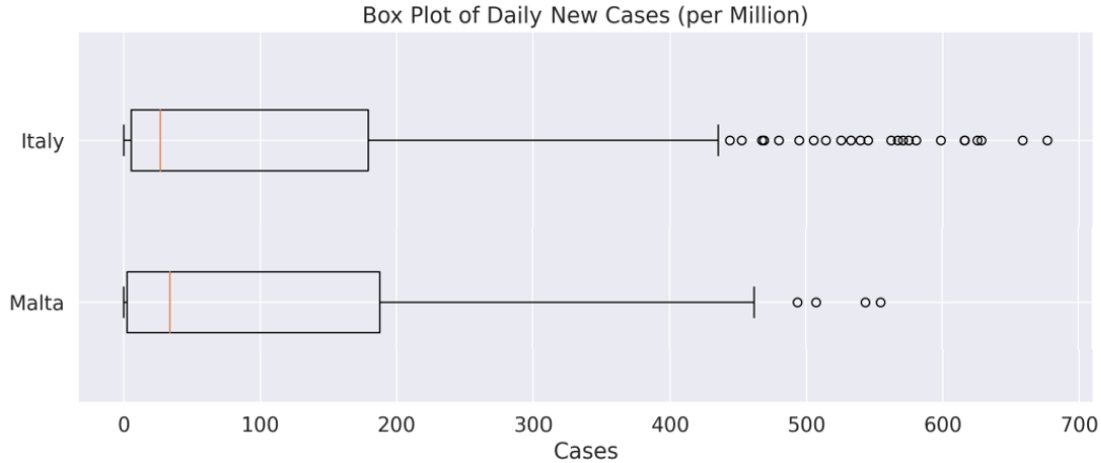

Figure 1.6: Box plot of daily new cases (per million)

Finally, new daily cases per million was used to visually compare the infection rates of Malta and Italy. The data is presented in 2 subplots, adhering to the small multiples technique. This prevents the messy overlapping of plotted data, also known as graph "spaghetti".

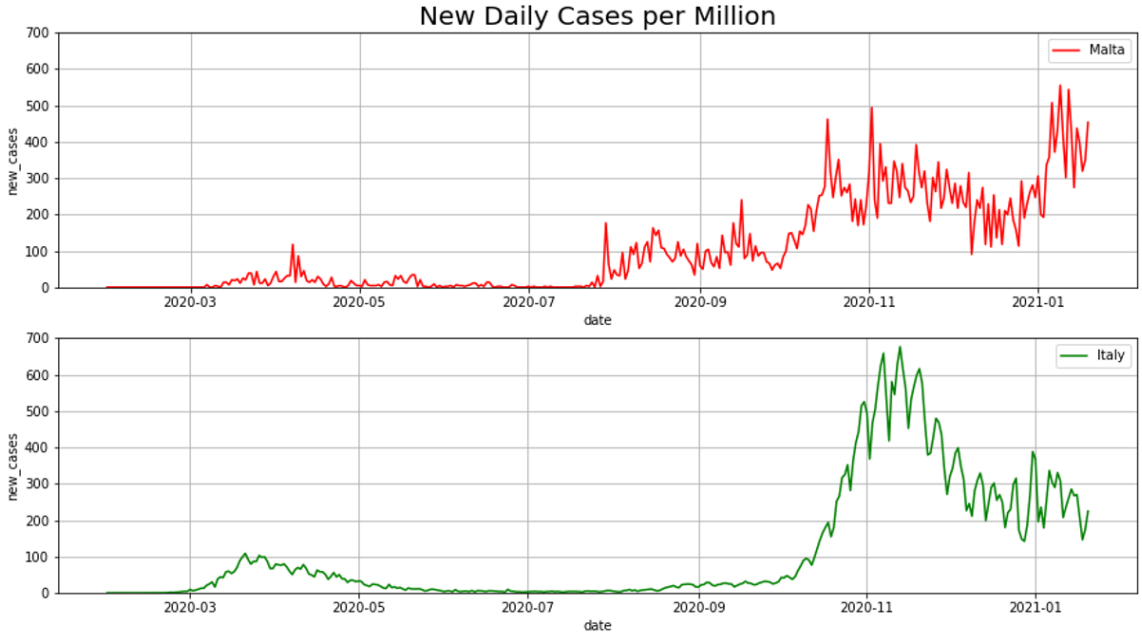

Figure 1.7: Line plot of new daily cases per million in Malta (top) and Italy (bottom)

1.2.2 Testing

Per capita, Malta has carried out 280% more testing than Italy in total. Because of this, Italy may be significantly underreporting cases, which would make the country's pandemic statistics in figure 1.5 underrepresentative. It is worth mentioning that cases are already underreported in general, since an entire country's population cannot be tested at once. Instead, a small sample of the population is tested daily. This mainly comprises symptomatic people who decide to get tested, or people in high-risk zones such as hospitals. We know that an infected (and therefore contagious) person can be completely asymptomatic, thus having no particular reason to get tested.

	Malta	Italy
Total PCR Tests (per capita)	1.31	0.49
Daily Max. PCR Tests (per million)	8925.6	4523.61

Figure 1.8: Total and peak testing done by Malta and Italy

1.2.3 Mortality Rate

Mortality rate is defined as *total number of deaths / total number of cases*. In the table, this is denoted by Mortality Rate (%) of Infected. Italy has the highest mortality rate (3.466%) whereas Malta has a significantly lower mortality rate of 1.501%. However, when taken per capita (or per million to avoid very small

numbers), the data shows that Italy has a significantly lower mortality rate than Malta. Here it is important to bear in mind that Malta is far more densely populated than Italy (1380 people/km^2 vs 206 people/km^2).

	Malta	Italy
Mortality Rate % (per capita)	3.39981e-06	5.73295e-08
Mortality Rate % (per million)	3.39981	0.0573295
Mortality Rate % of Infected	1.50115	3.46625

Figure 1.9: Mortality rates for Malta and Italy

1.2.4 Correlation between Testing, Infection and Mortality Rates

The correlation matrix for Malta shows a high positive correlation of 0.827 between testing rate and infection rate. There is a weaker correlation of 0.593 between testing and mortality rate. The correlation between infection rates and mortality rates is even lower (0.440). It is evident that the more testing carried out, the more cases emerge. This is why Italy's lower testing rates may be causing significant underreporting of cases. The population sample (tested individuals) becomes more representative of the true population (all individuals in country) as test coverage increases.

1.3 Relational Data Models vs NoSQL Data Models

Relational data models employ transactions whose properties can be described by the ACID acronym (Atomic, Consistent, Isolated, Durable). This ensures that each transaction is an indivisible unit of work that can either succeed or fail **entirely**, without partially applying changes. Furthermore, data must be in a consistent state before and after each transaction, which is paramount for any data science project. The database must only be subject to changes from one source at a time, and finally it must persist even in the event of a system crash. The latter is desirable for larger data science projects, but not necessary for the COVID-19 task since there are no lengthy computations being done on the data. Not all NoSQL data models are ACID compliant.

- **Structure**: Relational (SQL) databases are constructed by tables made up of rows and columns (schema), where type consistency is enforced. This is a desireable trait for all COVID-19 data collected, since it is difficult to make insertions without defining and adhering to respective datatypes. SQL databases allow us to build relationships between tables, and support powerful and complex queries which can access and retrieve records across many tables. NoSQL databases are implementation-dependent, meaning they can adopt several structures like JSON documents, tables or graphs. A relational database is definitely the more appropriate for the collected COVID-19 data. When used with a primary key such as date, any desired range of records satisfying complex constraints can be retrieved from multiple tables belonging to different countries, in one query.

- **Storage**: The data contained within a relational database is usually concentrated into a single file, and is typically not further partitioned or segmented. On the other hand, NoSQL databases make use of a hash function to return a value corresponding to an inputted key. The value is located in one of several nodes and then retrieved. Fundamentally, the NoSQL database relies on key-value storage. Therefore, without prior knowledge of the key, there is no guarantee that values can be efficiently

obtained. A relational model is more appropriate for running queries which will hop from one table to another or use a join operation to yield a new table with features of interest e.g, testing, infections and deaths.

- **Scalability**: The NoSQL database uses simple data models and scales extremely well compared to the relational database. Naturally, this come at the cost of sacrificing the query flexibility of relational databases. It tends to be the preferred choice for large scale projects with **known** database access patterns (otherwise SQL may be safer due to its flexible queries). Conversely, relational data models are more limited in terms of scalability and data reallocation can be a challenging task to undertake.
- **Access**: Interfacing with relational databases requires typing raw SQL syntax, whereas NoSQL alternatives make use of various REST APIs / CRUD syntax depending on vendor e.g., MongoDB / DynamoDB. The universality of SQL through all relational DBMSs, and the fact that it is thoroughly documented can be a reason to choose it over newer NoSQL systems.

In summary, the appropriatness of each data model depends on the data. If we are modeling e.g., Facebook connected friends, a graph-based NoSQL data model would most likely be ideal. Given the COVID-19 data, I feel that a relational data model is more suitable. The scalability disadvantage does not outweigh the ability to harness relations between country tables and run complex queries involving joins to return specific data of interest.

- **Critical evaluation**. The MySQL engine was implemented as a proof of concept but not utilized to its full extent due to time constraints. Instead, I relied more heavily on pandas dataframes which are essentially temporary variables. Setting *date* as the table primary key would have caught the outlier in figure 1.2 earlier on, which shows that the database does not just serve a storage purpose but also helps ensure data correctness.

2

Visualization and Statistical Analyses

2.1 Caïssabase - Visualizing 4 Million Chess Games

For this visualization task, the Caïssabase chess database was downloaded in SCID format, converted to Portable Game Notation Format (PGN) and parsed using the *pgn-parser 1.1.0* Python module. The database comprises an impressive 4.27 million chess games which took place all over the world through the years, with a few games dating all the way back to the 17th century! For starters, I felt it would be interesting to visualize where the chess games took place on a global or continental scale. Therefore, the first charts I generated were "bubble" maps showing how the database's chess games are geographically distributed. This type of visualization also compares the amount of games played in different locations in an easily interpretable form. This is because the bubble radius is directly proportional to the number of games played in a particular location.

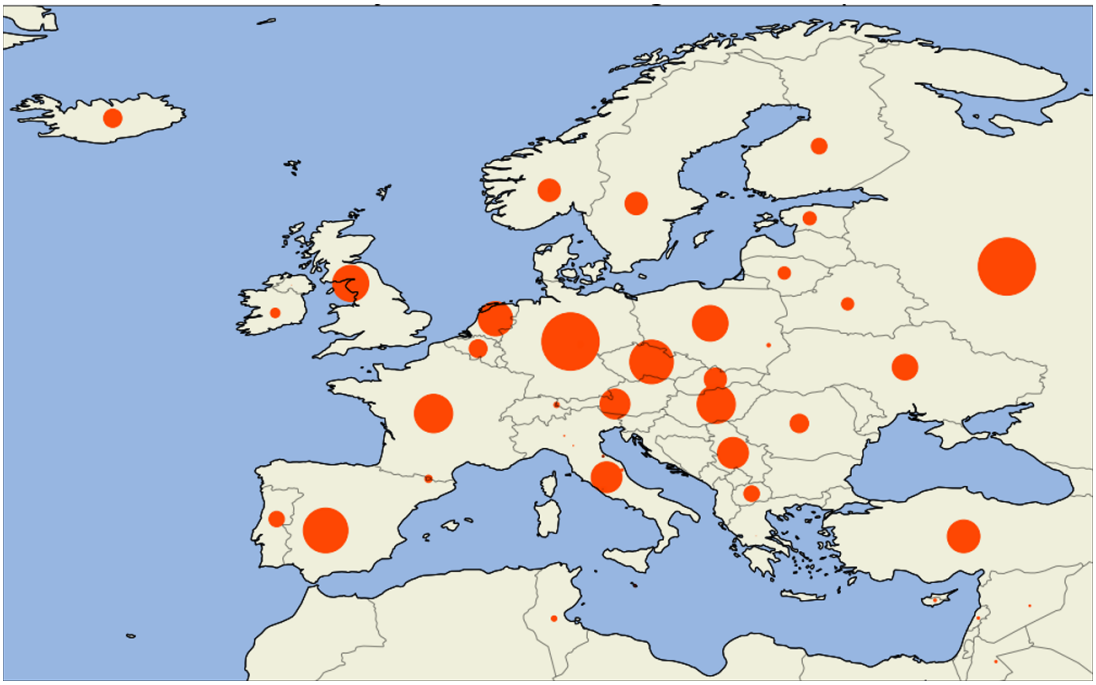

Figure 2.1: Bubble map of chess games played in European countries.

A similar chart was generated showing the games' geographical distribution on a global scale, which can be found in the *geomap.ipynb* notebook file. From figure 2.1 it is clear that Caïssabase has a good coverage of games played in Europe. More prominent countries include Germany, Russia and Spain. It is worth noting that each country is represented by **one** bubble. Also, Russa was taken to be entirely within Europe, when in reality it is a transcontinental country.

- **Critical evaluation**. Game location (site) is denoted inconsistently throughout the database. PGN specifies the use of International Olympic Comittee (IOC) country codes, but the database does not fully adhere to them. Some records only provide a city name. Furthermore, there are countries denoted by multiple codes. For instance, Germany appears as GER, DDR and BRD, the last two indicating East and West Germany respectively during the period 1945-1989. Also, some countries are subdivided into regions. For example, the UK appears as ENG, SCO and WAL. It is possible to catch such subdivisions and regroup them into their country, as done in *geomap.ipynb*. However, game sites which appear as city names such as "Bremen" or "Paris" without any uppercase country code are not included in the visualization.

Next, two bar charts were generated to visualize the lethality of each piece type. Figure 2.2 shows the number of captures made by each of the 6 piece types.

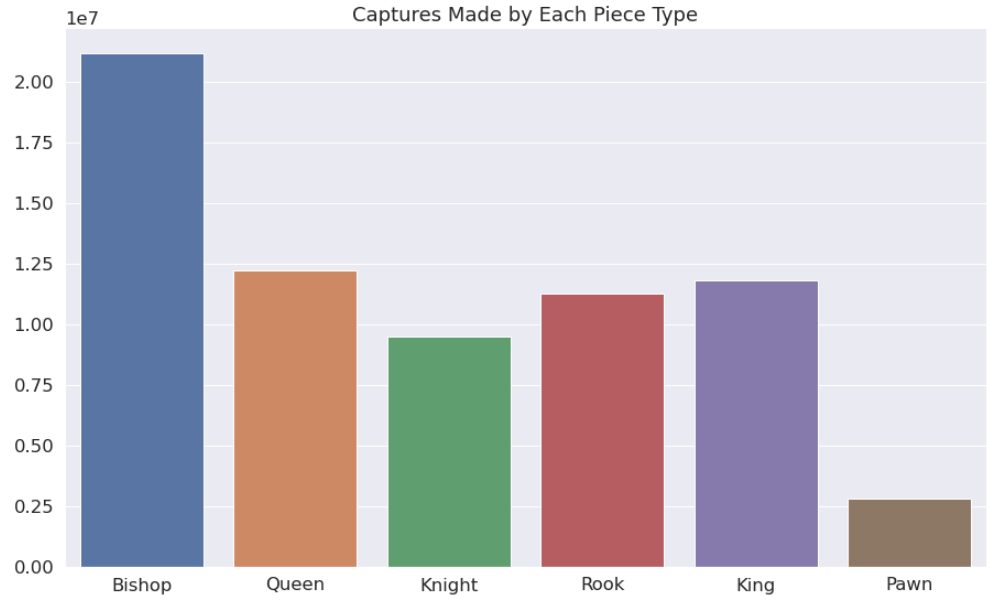

Figure 2.2: Bar chart showing number of captures by piece type.

Clearly, most captures are made by bishops, which can be said to have the highest "lethality". However, this visualization is really measuring **how aggressively each piece type is used**, since there is no such thing as piece lethality but rather how its movement qualities can be best harnessed in order to win. The low number of captures by pawns may seem counter-intuitive since a player has 8 pawns at their disposal. However, this is far outweighed by the pawn's limited range of movement, as it can only capture a piece situated **one** square diagonally **ahead**.

The second bar chart was obtained by normalizing per piece. For example, total captures made by bishops are divided by 2 since a player has 2, whereas pawns are divided by 8.

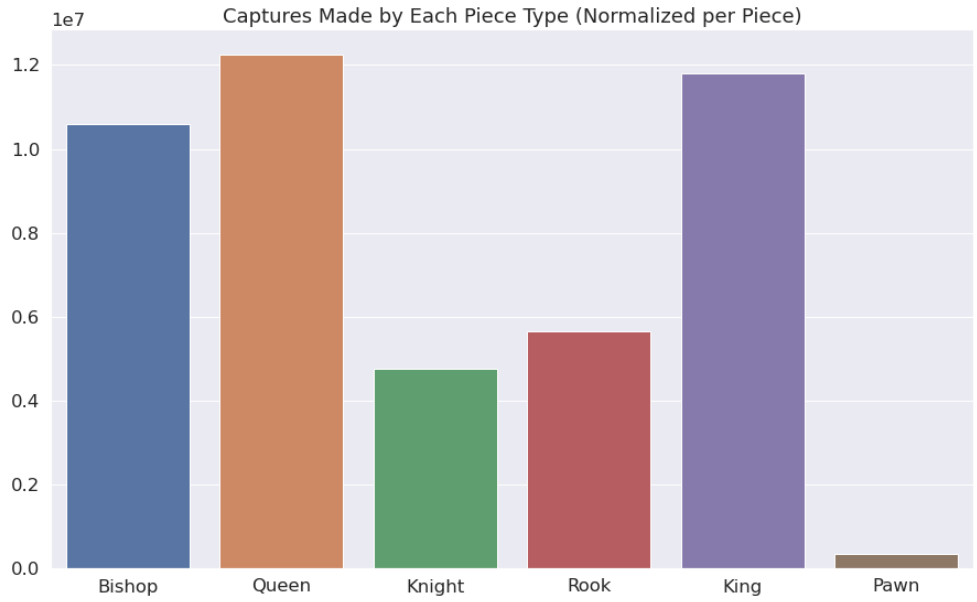

Figure 2.3: Bar chart showing number of captures by piece type normalized per piece.

Per piece, the king and queen are now observed to have made more captures than the bishop, and each pawn 8 times less than before. While this visualization is not directly representative of a real chess game, it still provides useful insights by illustrating the "value" of each individual piece. Suppose a player finds themselves in a non-check situation where multiple pieces are threatened, and they must choose which one to save. Losing one pice may be worse than losing another. The chart **suggests** that in such a situation, a player chooses a rook over a knight, or the queen over a bishop etc.

Next, a two-dimensional heatmap of the most dangerous squares was generated.

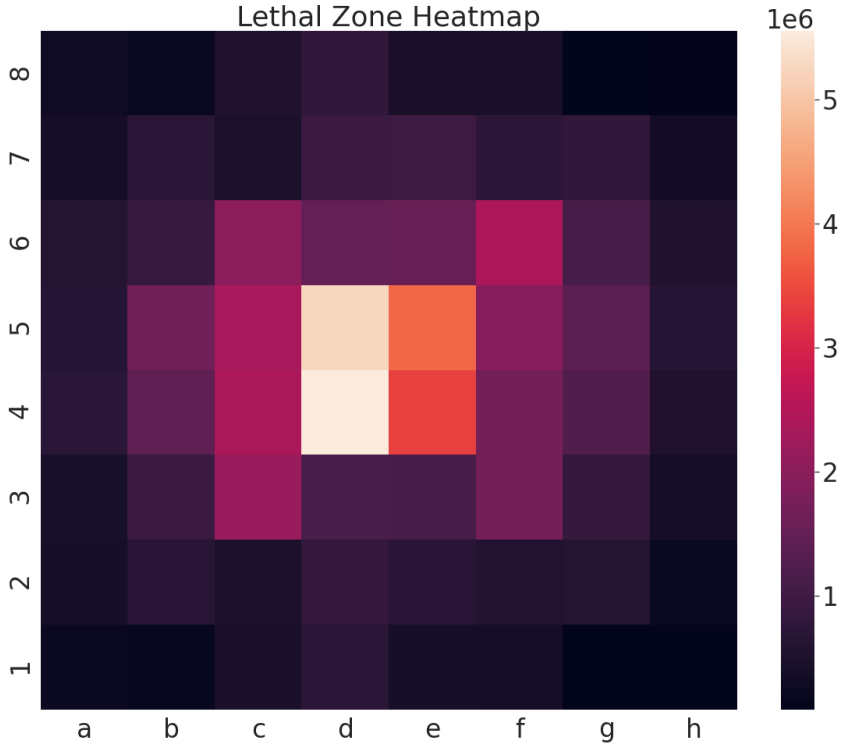

Figure 2.4: 2D Heatmap showing where pieces are most likely to be captured.

The heatmap excels at descibing many games on a single chess board. In my case, the heatmap shows the frequency of captures for each of the 64 sqaures in an easily comparable way. The reuslts are roughly as I expected, with the least amount of lethalities occurring on the chessboard perimeter. Capture probability increases as pieces advance towards the centre of the board, and the 4 central squares {*d4, e4, d5, e5*} are shown to be the most dangerous, since piece captures are concentrated within that area. There is also a slight lethality bias to one side of the central area demarkated by {*c4, d4, c5, d5*} i.e., it is more lethal than the area {*e4, f4, e5, f5*}. Also, capture location does not appear to be skewed towards the white or black starting positions.

Finally, a histogram was generated to visualize how the game duration is distributed. Since PGN does not include game time, I chose to represent the length of a game using the number of moves per game. The histogram shows that the range of moves per game extends from around 10 moves to well over 500.

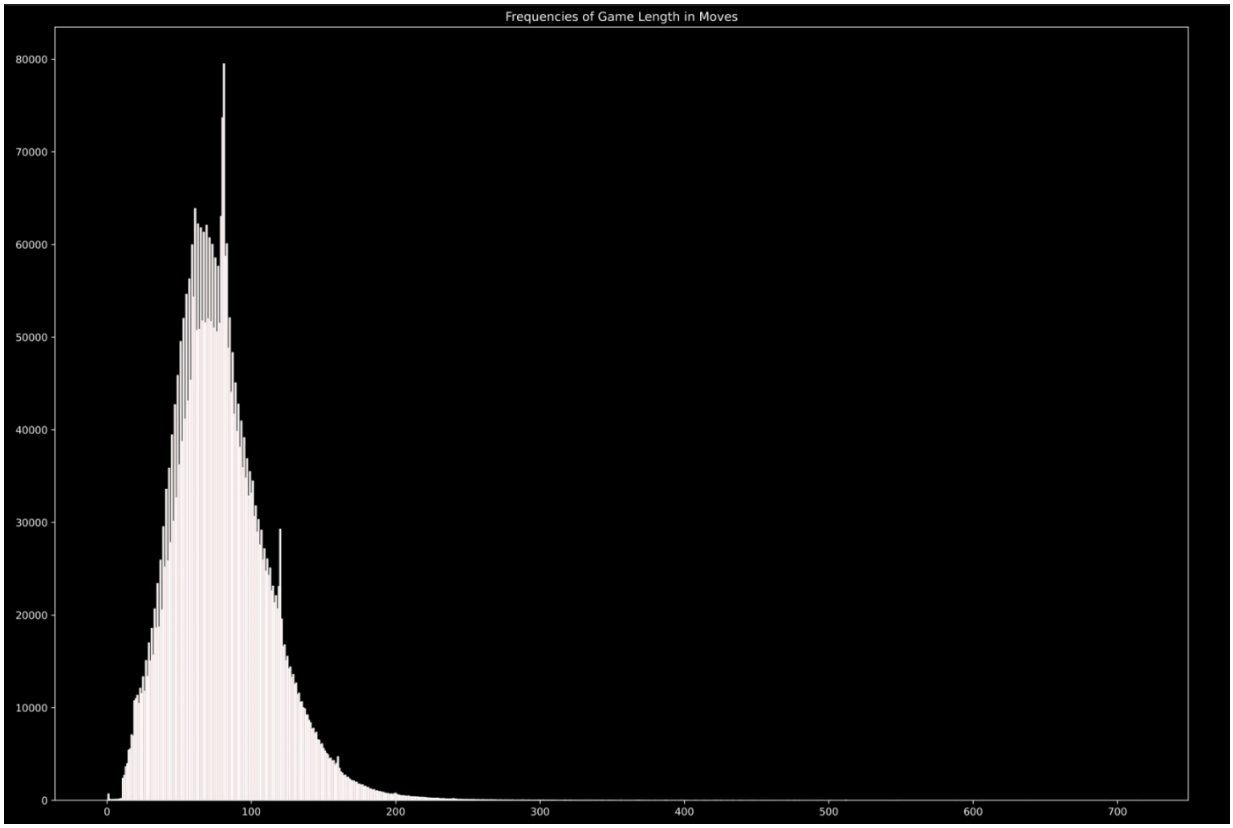

Figure 2.5: Histogram denoting number of moves per game

The games with the fewest amount of moves are likely the outcome of mismatched players in terms of skill, where a less experienced player is not likely to last long against a seasoned player. Histograms are useful for identifying outliers. In fact, the plot clearly is **positively skewed**, and this is caused by outlier games which had unusually large numbers of moves.

Three spikes are observed in the histogram at approximately 100, 120 and 160 moves. It is not entirely clear what is causing these spikes to occur, however this might be a consequence of **time control**. Time control plays a central part in chess, in fact **most** chess games are timed. For example, in "5-minute Blitz" each player has a timer which is initially set to 5 minutes. Their timer counts down whenever it is their turn

to make a move, forcing them to think quickly. The timed nature of a game likely impacts the resulting number of moves, since a player has less time to figure out an optimal move. There are several other forms of time control in chess, but PGN does not specify the protocol used in each game. Hence, the spikes may reflect different timing protocols within the database.

2.2 Identifying and Fixing a Bad Visualization from Local Media

The following visualization was identified from the local media[1]. It shows the results of a webpage community poll about the best places to get a burger in Malta and Gozo. Though the intentions behind it are great, it is a bad visualization for several reasons.

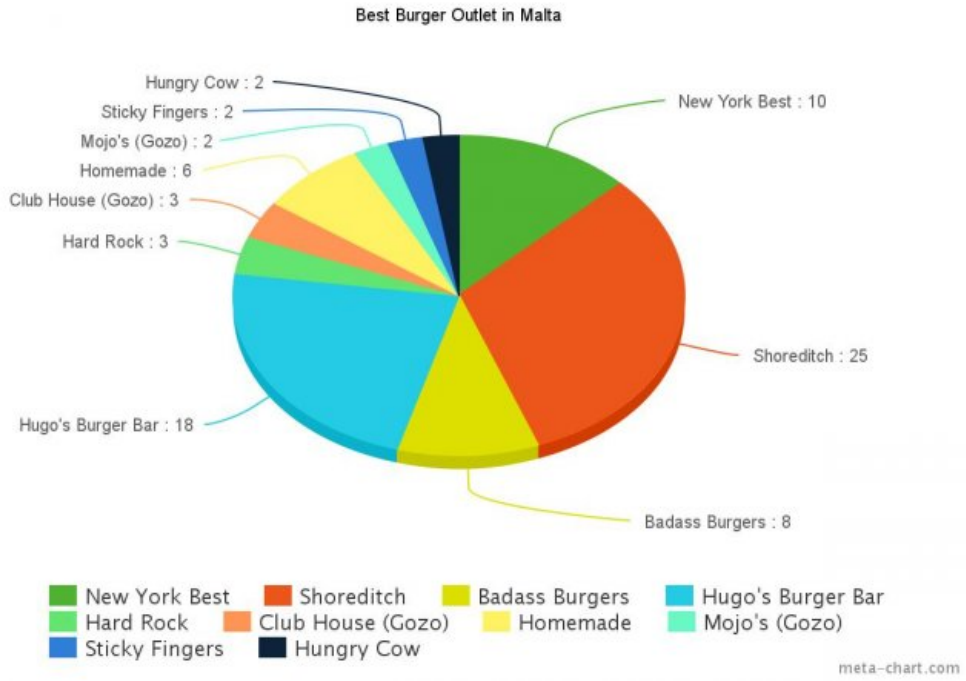

Figure 2.6: A bad visualization found in the local media.

- **It is a pie chart**. The human visual system (HVS) performs poorly at judging relative proportions and areas of segmented areas, such as pie slices. Also, the pie chart's segments are not ordered by size, making interpretation harder.
- **It uses three dimensions unnecessarily**. There is no reason to use more dimensions than required by the data. Here there are two dimensions: *restaurant* and *votes*. Showing the same 3D chart from different angles will make certain parts appear larger or smaller than others, as the HVS tries to correct for 3D objects being mapped onto a 2D plane. For instance, the segment corresponding to *Badass Burgers* makes up more than its true 10.13% of the visualization since everything is tilted.
- **There is cluttering of labels around the pie chart**. The lines stemming out of each segment are cluttered between the 9 and 12 o'clock positions. This further contributes to a less interpretable visualization.

[1]https://www.swag.com.mt/en/articles/articles/770/vox-pop-wheres-your-favourite-burger-from.htm

- **Use of similar colors**. Since the chart uses very similar colors, it is difficult to discriminate between the different categories. For intance, it is easy to get confused which shade of green belongs to which category, resulting in back-and-forth legend consultation.
- **There is redundant duplication of information**. The legend is effectively duplicated around the pie chart. This is possibly an attempt to eliminate the ambiguity introduced with using similar colors.

The visualization was reimplemented as a bar chart, where outlets are ordered in descending popularity. Bar color harnesses pre-attentive vision to quickly distinguish between outlets in Malta and outlets in Gozo. Also, each bar's value is positioned just above the bar for easy interpretation, since there is some distance between the rightmost bar and the y-axis.

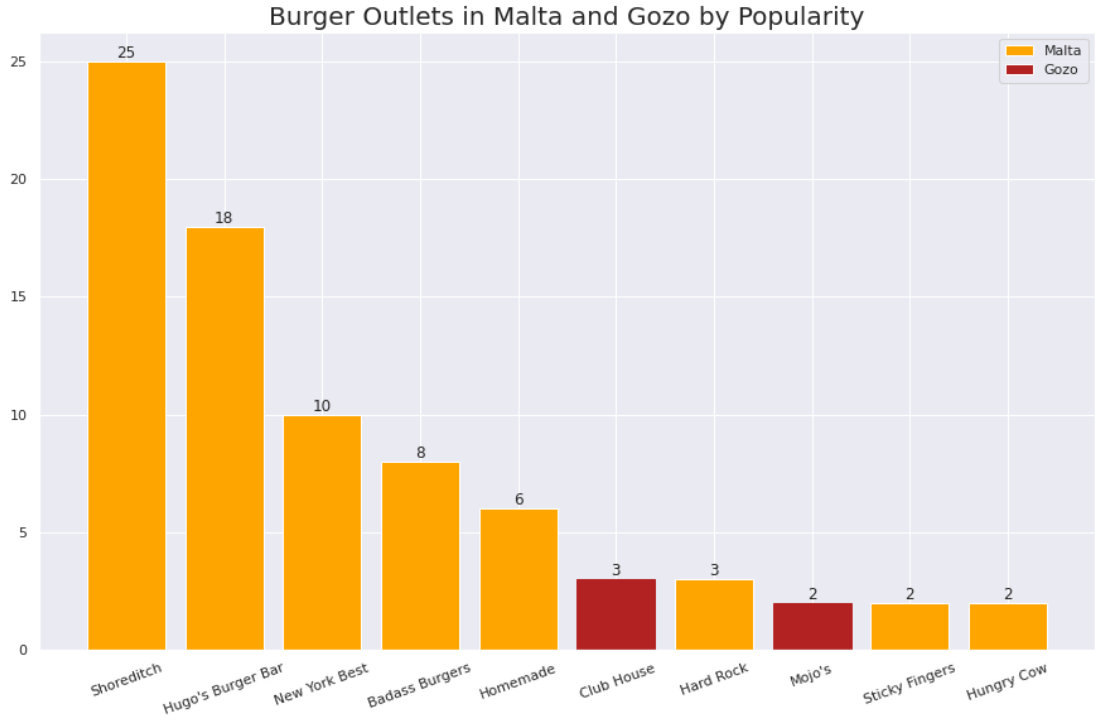

Figure 2.7: Reimplementation of 3D pie chart as 2D bar chart.

3

Data Science Project - Dataset Analysis

3.1 Discussing the Supplied Data

There are 480 files within the *202021_CPS3235_data.tgz* archive file, half of which are *html* files belonging to Times of Malta's *classifieds* property webpage. The remaining 240 files are *wget* log files showing download information such as date, time and status. Aspects of the data in the *html* files are discussed:

- There are several types of data provided. The *property type* is an example of **qualitative categorical (nominal)** data e.g., "maisonette", "apartment" or "penthouse". It is nominal because there is no implicit ordering of the categories. *Location* and *contact number* are also examples of qualitative categorical data. Despite being numeric, there is no implicit ordering of phone numbers, and it makes no sense to find the "mean" phone number. *Price* is a good example of **quantitative continuous** data, as its value can be any real number although we typically round it to the nearest hundreth, depending on the scenario. There is a short description, typically one sentence, describing the property. Keywords like "bay views", "own roof", "quiet area" and "terrace" are further examples of categorical data, which can take *yes/no* values.

- The data is directly obtained from the source, namely the Times of Malta's (ToM) *classifieds* website. The collection method can be considered an automated survey involving the selection of a feature (properties for sale) taken from a population sample. In this case, the ToM classifieds could be a sampling frame of the national real estate market. Alternatively, the whole population could be ToM property listings if the scope of analysis is limited to ToM's property advertising. The log files indicate the use of systematic sampling, as new data is retrieved in weekly intervals from April 2015 to January 2020, with a couple of exceptions. However, despite there being a sampling strategy, **all** listings are still obtained, since the webpage is updated daily and includes listings backdated by a week. So if one considers the whole population to be property listings on ToM, the entire population data is obtained.

- The data does not follow a strict format. Although the vast majority of listings are selling individual properties, ads for for companies can be found where no price, property type or specific description is availble. Furthermore, the *html* layout of the website was observed to change slightly at some point, though the data was unaffected by this.

- With regard to timeliness, the data spans from mid 2015 to early 2020. Therefore, it can be considered up-to-date or not, depending on what it shall be used for. The real estate market is known to

fluctuate throughout the seasons, as holiday season and scholastic period both influence the supply and demand of real estate. However, the local market may be less prone to fluctuations compared to other foreign real estate markets. Nonetheless, it is beneficial that there is data spanning throughout all seasons.

- The data is largely assumed to be correct. Some obviously incorrect entries containing meaningless strings are observed throughout the data which should be discarded. Apart from the latter, the data is generally correct because ads are listed to **sell**. It is in the advertiser's interest to ensure their listing contains correct data since they are most likely paying to list their property on ToM. Also, they have no reason to be deceptive with property type, price and location as this would be counterproductive from a selling point of view.
- In terms of completeness, there seems to be data available for every day with the exception of a few cases. For instance, there is no data between April 30th and May 4th. As mentioned earlier, there is generally an overlapping day between 2 consecutive files. The first day in a file is included as the last day within the next file. This is a duplication of data and will need to be handled appropriately during collection and cleaning. One should bear in mind that this data is solely from ToM, and may not be representative of all property for sale in Malta. For example, maybe ToM's demographic is known to be more interested in houses than villas, and so less people advertise their villas on ToM.
- I believe that the data is reliable. Nobody pays to list their advertisements just for the thrill of it! As soon as a propery is sold, it is in the seller's interest to remove the listing. Therefore, I would confidently say that all ads are/have been tangible real estate opportunities, and there are very few (if any) lingering listings that are falsely listed as for sale when are in fact sold.
- Finally, in terms of relevance, every file contains site header and footer data. This mainly consists of links to other ToM pages like sports, careers, popular stories, and login forms, all of which are **irrelevant** to the task at hand.

3.2 Selecting and Justifying Features of Interest

Obvious features of interest which appear consistently throughout the data are:

- Property Type
- Location
- Price

However, other features such as area in square meters, or pool/garden space can be considered too. The collection of these features would have allowed for some interesting investigations, but I noticed they were included extremely inconsistently in listings. Therefore, I would most likely not be able to arrive at such solid conclusions. I decided to solely analyze property type, location and price due to their consistent inclusion in the data.

Beautifulsoup was used to parse the html found in each of the 240 files. It was observed that each property listing's text is contained within a list item <*li*> tag. These tags were also noted to have a *name* attribute which is unique to each listing. This attribute proved to be particularly useful since overlapping listings in two consecutive files have the same *name* attribute. I was able to exploit this to get rid of all duplicate data, simply by calling *pandas.drop_duplicates()*. Therefore, the *name* attribute of each <*li*> tag can also be considered a feature of interest since it is crucial to collect.

After converting everything into lowercase for consistency, regex was used to search for municipality (location) by exploiting the fact that these consistently occurred after a line terminator and generally were not longer than 30 characters. There were varying levels of specicifity across locations, which needed to be normalized. Therefore, matched locations were checked for single keywords and overwritten with a locality. For instance, the string "mellieha heights" would be snapped to "mellieha" since it contains the substring "mellieha". Checks were also implemented to catch common misspellings of maltese town names. It is possible that some locations got errouneously omitted due to such checks not being adequatetly exhaustive. Nonetheless, a good amount of locations were successfully extracted. Note that Gozo was taken to be a single location for this task.

A similar pattern-matching technique was used to extract property and price. Each feature of interest (including <*li*> *name* attribute) was stored in an independent list, so it is crucial that the list elements correspond to each other in the correct sequence. This was ensured by always adding **one** new item to each list per <*li*> tag parsed. If no regex match is found, then *np.nan* is appended.

After the data collection was complete, the NaN percentages were calculated for price, location and property type. They were found to be 22%, 39% and 10% respectively. From these, price is the most practical to impute. Taking the median of all prices is a naive approach, as this will assign the median price regardless of property type. For example, a villa will get an imputed price of 245k euros, which is a severe underestimation of what a villa should cost. Therefore, I take the median price per property type, and perform imputation based on the property type. One could take this a step further and find the median price for all property type and location combinations.

3.3 Extracting Knowledge From the Data

The knowledge extracted form the data should be useful to the property market. To this end, I have considered four scenarios, namely.

- A first-time buyer looking for an affordable apartment.
- An investor in search of an expensive villa.
- A first-time buyer in search of an entry-level maisonette.
- A property investor with an interest in a luxurious aparment.

The aim is to identify the most suitable localities to satisfy the requirements of each of the four.

A first-time buyer looking for an affordable apartment
The data shows that *Zabbar*, *Gozo*, *Qormi*, *St Venera* and *Bormla* are the top 5 cheapest locations for apartments on average.

An investor in search of an expensive villa
The data shows that they should search for villas *Gharghur*, *Bahar ic-Caghaq*, *Mellieha*, *Ibrag*, and *Birgu*. These are the top 5 most expensive locations for villas.

A first-time buyer in search of an entry-level maisonette The data shows that the cheapest maisonettes are located in *Bormla* with an average price of €161,530. Following are *Isla* (€165,848) and *St Venera* (€173,001).

A property investor with an interest in luxurious aparments The data shows that they should search for apartments in *Qawra*, *Birgu*, *Valletta*, *Sliema* and *Isla*. These are the top 5 most expensive locations for apartments.

Regardless of the type of buyer, it is important to examine the most frequently advertized properties by type. The predominance of apartments on the market may not simply reflect demand, but may also be a consequence of the relatively recent legislation allowing for the addition of storeys to existing property. In broad terms, the bar chart shows, with a few exceptions, that supply is inversely proportional to cost of property. Among the exceptions are fields and penthouses. In both cases, this is because their location is severely restricted in that they are either beneath the building (fields) or at the top of the building (penthoiuse). This is in contrast to apartments and maisonettes which may be distributed throughout a larger single building. Maisonettes also consitute a slight exception, which is likely to be a result of their being more closely priced to houses rather than apartments.

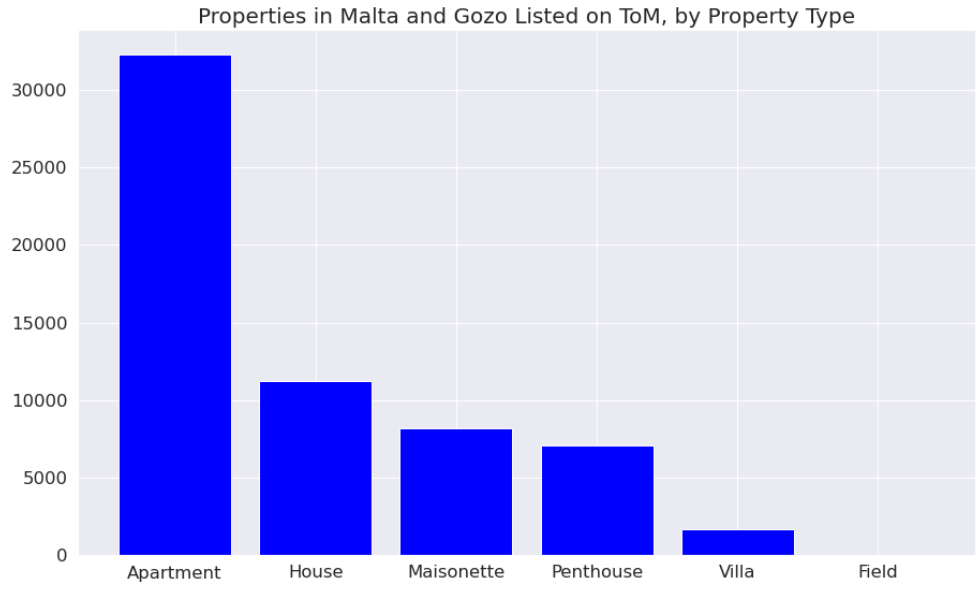

Figure 3.1: Most frequent property types for sale on ToM

3.4 Building a Predictive Model

The linear regressor was chosen to predict property price based on location. This model uses the **least-squares** method to find a line of best fit for the scattered data, and a **coeffiecient of determination** R^2 to denote how well the model performs.

- $y = b_0 + b_1x_1 + b_2x_2 + ... + b_nx_n$

Where y is the dependent variable (price), b_0 is the y-intercept, x is the independent variable (location), $b_1, b_2, ..., b_n$ are the slope coefficients, and n is the number of observations.

The model is typically used with continuous data, however in this case it is mapping to the set of real numbers based on a group of categorical variables. Therefore, some form of encoding is required since linear regression cannot work with categorical data directly. Simple *integer encoding* would have been a viable solution had the independent variable been ordinal data i.e., followed some implicit ordering. However, this does not hold true for categorical data as there is no relationship between locations, thus a different approach is required. The solution to this problem is *one-hot encoding*, which splits the location column into several columns containing binary variables, also known as dummy variables. A common pitfall is associated with one-hot encoding is the introduction of multicolinearity within the dataset. The standard practice to avoid this is to drop one dummy variable column.

The 10 most frequently occurring locations were considered for the linear regressor. Also, the collected data was split into two sets, one for training the model and another for testing it. This is necessary because if a model is evaluated with data it has already "seen" during training, then its true ability to generalize is **not** being measured. The training and testing sets were split in the ratio 80:20, prioritizing the training data. When evaluating the model, a negative R^2 value of around -0.000138 was obtained. This indicates that given the imposed constraints, the model has fitted the data poorly.