HMO Caps in St Andrews
2022 Martinmas Semester Progress Report

By
Akshitha Vemuru, Ann Kathryn Chaney, Aoife Doyle, Ben De Mello, Georgina Orchard, Kira Siebrecht, Makenna Hartwich, and Qianyu Zhang

Submitted for Review: 31 October 2022

TABLE OF CONTENTS

I. Introduction

This semester, the project goal is to *“quantify the impact of the HMO License Cap on the St Andrews rental market using real-world data”*. Following the recommendations and next steps of the previous groups, we have decided it is the right time to begin the collection of real-world data. This semester our efforts have been concentrated on finding appropriate secondary data sources, analyzing this data and improving the economic model created last year. The overall aim of developing this economic model is to prepare the model for the data collected in a survey that will be conducted during semester two.

This progress report will highlight the work that has been completed by the current project group from weeks 1-7 of Martinmas Semester, 2022-23. Significant milestones include the creation and analysis of a HMO database, web scrapping rental properties in St Andrews and creation of a Survey, ready for launch in semester two.

The remainder of the report will be divided as such: Section II will discuss the literature that has been reviewed. Section III will discuss the data collection, manipulation and analysis from Fife Councils’ HMO Register. Section IV presents the results of the data analysis conducted on the HMO Register, relating these back to the overall project goal. Section V will discuss the process of web scraping rental prices over time in St Andrews and Section VI presents the results and correct interpretation of our findings. Section VII presents the start of the survey design and ethics application. Section VIII details the survey process. Section IX discusses the intended structure of the survey. Section X reviews general lessons learned on team workflow. Lastly, Section XI outlines the next steps for the project for the remainder of the semester and beyond.

II. Literature Review

Housing Need and Housing Pressures in St Andrews (2013) by St Andrews Town Commission on Housing

This resource provides historical context on the HMO policy and its effect on the St Andrews rental market. The authors introduce the concept of studentification, which we aim to measure in our future work using the data collected and explained in this report.

The Rapidity of Studentification and Population Change: There Goes the (Student)Hood (Sagel, Smith and Hubbard, 2012)

This publication provides a formal definition of studentification as a contentious social issue. Authors explain that previous literature surrounding the 'studentification' issue has failed to provide explanation for the growing gentrification due to the in-movements of young, middle-class populations to residential neighbors in university towns; and go on to provide their own account of the increasingly visible process that is occurring.

Growing Economies and Building Houses: Reconciling Growth and Housing Wellbeing in St Andrews (Maclennan, O'Sullivan, Maynard, Sila-Nowicka, Walden, 2013)

This 2012 survey helped provide inspiration to our own project pathway. This study was based on three local surveys and presented a concise analysis of the University staff and students location choices from administrative data and interviews with market agents and investors. They investigate the problems arising in the St Andrews Housing Market. We can look at their conclusions and determine whether our preliminary data and future survey responses have common features. If not, then it will be a useful analysis in explaining the differences in the St Andrews housing market since 2012.

III. HMO Register Data

What are we trying to get from our data?

We identified the HMO register database as an important secondary data source that we could use to conduct longitudinal analysis on trends in the applications of HMO licences in St Andrews over a long time period, including the years of the HMO freeze. We decided that looking at the number of applications of HMO licences each year would be important for checking that the issuance of the HMO freeze was followed and if there were any significant trends that occurred over these years. With the information in the register, we also decided that looking at the number of 'available bedrooms' each year would be significant in looking at trends within the data. When considering the economic model of supply and demand of rental properties in St Andrews, we concluded that the HMO register data could be interpreted as the supply of rental properties.

During Week 1-7, we have focused on extracting the exact number of properties and as a result number of bedrooms where HMO licences have been issued. As the register has data before and after the HMO freeze (pre-2019 and post-2019) this allows us to evaluate if the HMO freeze has restricted the supply of rental properties in St Andrews.

Methodology of Data collection

As part of the Housing (Scotland) Act 2006 Part 5, Fife Council are required to keep a register containing information regarding HMO licences for living accommodation situated in its' council area and information on applications for these licences. We obtained the most up-to-date pdf of this register, published in Q1 2022 from Fife Councils' website.[1] The HMO register contained all licences applied for between the time period of 2004 and the first quarter of 2022.[2]

Due to the familiarity of programming languages within the group conducting the HMO register analysis, we decided to use R, as it is particularly efficient at statistical computing. The team carried out all coding activity in RStudio, an integrated development environment for R.

The register published on the Fife Council website, in the form of a PDF, could not be easily imported as a dataset into R. After the first attempt, we decided that it would be easier to convert the PDF to an Excel spreadsheet and then read the Excel file into RStudio. The full code of analysing the HMO register is shown in Appendix 1.

There were several packages and libraries that needed to be imported to read the Excel file and conduct further data manipulation and analysis. These extend R's capabilities through user-created packages allowing the free use of others' tools. The below table explains the essential packages we used in the code.

[1] https://www.fife.gov.uk/kb/docs/articles/housing/private-rented-sector/homes-in-multiple-occupancy-licence/hmo-public-register

Figure 1

Package & Library	Description
tidyverse	Tidyverse is a package that makes it easier for the human coding to communicate with the computer. It contains packages which provide functionality to model, transform and visualise data easily.
readxl	Reads an Excel file into R.
dplyr	Package that allows essential shortcuts for fast data manipulation.
sampling	Allows user to obtain different estimators of datasets. Also allows user to draw random samples from both real and simulated datasets.
writexl	Creates an Excel file from a data frame in R.
lubridate	Package that compliments 'tidyverse' expanding mathematical operations with date-time objects.

The code for tidying the HMO register consists of three parts. The first part imports the HMO register from Excel and using R we tidied it up into a form so we could begin the data analysis. The second part reads the cleaned file back into R and we wrote a loop to call each year of licences issued as a separate dataset. The third part is a script that allows the user to enter a specific time period and R returns a data frame including all properties who had licences issued within the given time period. This is what we refer to as *'active licences'*.

Due to the formatting differences between the PDF, Excel and RStudio we initially had a very large database with multiple empty rows and columns, so we eliminated any rows that had no data in them, or rows with no data for the '(Applicants)Name' as an inspection of these rows found that they were an implication of passing the dataset through different software tools rather than containing relevant information. We renamed the column names for ease in the following lines of code.
Next, we omitted any datapoints where the address was not in St Andrews. We used conditions that rejected any data entries with postcodes that were not the postal code, KY16, for St Andrews and surrounding towns. To eliminate the surrounding towns of St Andrews such as Leuchars and Guardbridge we searched the dataset for keywords and rejected these entries.
After conducting all relevant tidying up of the initial PDF HMO register, we are left with 6440 total relevant data points.

After manual inspection of the dataset named "`df`", we saw that there were duplicate entries of the same properties but under different names for the same year. As we wanted to keep all separate applications of HMO licences over the time period but make sure that each application was only counted once, we couldn't use the code below as this removed all duplicated addresses regardless of the year the licence was issued.

```
data[which(duplicated(data$Address)==T),]
```

We identified conditional formatting in Excel as a more accurate way to sort through this issue, so we exported the dataset to Excel to remove these. In this Excel file we split the dataset into years by the year of issuance of the HMO licence. Within each sheet we applied a conditional formatting rule which highlighted duplicate rows by their address. We eliminated duplicate rows, leaving us with just one entry per address.

After conducting this formatting we were left with a total of 4918 entries. We also removed the columns named ‘Number’, ‘Name’, and ‘Ward’ as these were not relevant for the next stages of data manipulation. We then read the modified Excel file back into R to continue with the data analysis.

The next stage was to determine exactly how many licences were issued in each year. We added an extra column to just have the year of issuance rather than a full time-stamp.

```
data$year <- as.POSIXlt(data$Issued)$year +1900
```

R takes the initial year in `as.POSIXlt(...)` as 1904 so we added 1900 to this command to get our start year to be 2004 as required.

We then created an empty data frame ‘`IssuedYear`’ which the execution of the loop creates a data frame for each year and stores it within IssuedYear.

The next part of the code allows us to filter the database by year of licence issued and the number of occupants (bedrooms).

Equation 1

```
library(dplyr)
data %>%
  filter(data$year==2004, data$Occupants==3)
```

We then wrote code that would allow us to determine exactly how many active HMOs there are at a given date.

Equation 2

```
#Created a script for searching for the number of active licences during a
time period.
data$Applied = as.Date(data$Applied, "%Y-%m-%d")
data$Issued = as.Date(data$Issued, "%Y-%m-%d")
data$Expiry = as.Date(data$Expiry, "%Y-%m-%d")
data_subset <- subset(data, Issued>= "2017-01-01" & Issued<= "2017-02-01")
```

The final line of code saves the set of results into a data frame given the condition of dates. In the example above, this creates a dataset where all entries that were issued between 1st January 2017 and 1st February 2017 can be viewed. The figure below shows the result of this command.

Figure 2

	Address	Issued	Expiry	Occupants	year
1	16a Greyfriars Garden St Andrews KY16 9HG	2017-01-20	2019-10-29	5	2017
2	2 Nelson Street St Andrews Fife KY16 8AJ	2017-01-12	2019-04-22	4	2017
3	181 South Street St Andrews KY16 9EE	2017-01-31	2019-08-29	4	2017
4	9 Crails Lane St Andrews Fife KY16 9NR	2017-01-10	2020-01-10	3	2017
5	10 King Street St Andrews Fife KY16 8JQ	2017-01-12	2019-10-21	4	2017
6	3B Playfair Terrace St Andrews Fife KY16 9HX	2017-01-10	2019-01-07	4	2017
7	3 Condie Court Market Street St Andrews KY16 9PD	2017-01-12	2019-05-20	3	2017
8	2 Claybraes St Andrews KY16 8RS	2017-01-13	2019-10-03	5	2017
9	76 Bridge Street St Andrews KY16 8AA	2017-01-30	2019-10-25	4	2017
10	7 Abbey Court St Andrews KY16 9TL	2017-01-10	2019-08-30	4	2017

IV. HMO Register Results

Methodology of Results of Database

To utilize the cleaned up HMO database for extracting trends through models, another code was written as can be seen in Appendix 2. First, the Excel file[3] with the cleaned up HMO licenses split into years by the year of issuance was imported. This dataset was what was used to compile all the tables and graphs.

After installing the essential packages, there was one address that still did not have an expiration date, as the date of issue was very recent and Fife Council most likely had not updated the given address' license expiration date. As a result, the call was made to put the expiration date as "2022-12-30". As the current research uses information up until 2022, and the average duration of a licence is over two years, this was a valid assumption to make. The figure below displays the change that was made.

Figure 3

	HMO Address	License Status	Date of Ap	Date Issued	Expire Date	Tot Occs	Decision	...8	Duplicated HMO Address?	year	yearrange
1	10 Greyfriars Garden St Andrews Fife KY16 9HG	Licence Issued	2020-11-09	2021-12-13	2022-12-30	4	XXWITH	NA	1	2021	2

To find the annual number of active licenses per year, a for-loop was used. As shown below, a table was called "hmo2" was created. The loop took the year of each licence's issue date and stored it in the variable "year". Then it found the number of years each license was active and stored it as "yearrange". The license got duplicated the amount of times stated on the "yearrange" column as can be seen in the table below the program. This way, each license is counted for the issue year, as well as every other year after because of the duplicate numbers that are the same as "yearrange".

Equation 3

```
#Finds number of active licenses per year
rows <- c()
years <- c()
for (i in 1:nrow(hmo)){
  rows <- append(rows, rep(i, hmo$yearrange[i]))
  newyear <- hmo$year[i] + 0:(hmo$yearrange[i]-1)
  years <- append(years, newyear)
}
hmo2 <- hmo[rows, ]
hmo2$year <- years
```

Figure 4

	HMO Address	License Status	Date of Ap	Date Issued	Expire Date	Tot Occs	Decision	...8	Duplicated HMO Address?	year	yearrange
1	5 Shorehead St Andrews Fife KY16 9RG	Licence Expired	2005-09-09	2004-06-08	2005-09-11	4	N/A	NA	0	2004	2
2	5 Shorehead St Andrews Fife KY16 9RG	Licence Expired	2005-09-09	2004-06-08	2005-09-11	4	N/A	NA	0	2005	2
3	1 Albany Park St Mary Street St Andrews KY16 8BP	Licence Expired	2011-05-11	2011-12-21	2014-05-15	6	GRANT	NA	1	2011	4
4	1 Albany Park St Mary Street St Andrews KY16 8BP	Licence Expired	2011-05-11	2011-12-21	2014-05-15	6	GRANT	NA	1	2012	4
5	1 Albany Park St Mary Street St Andrews KY16 8BP	Licence Expired	2011-05-11	2011-12-21	2014-05-15	6	GRANT	NA	1	2013	4
6	1 Albany Park St Mary Street St Andrews KY16 8BP	Licence Expired	2011-05-11	2011-12-21	2014-05-15	6	GRANT	NA	1	2014	4
7	1 Albany Park St Mary Street St Andrews KY16 8BP	Licence Expired	2014-04-14	2015-04-07	2017-05-15	6	CON	NA	1	2015	3
8	1 Albany Park St Mary Street St Andrews KY16 8BP	Licence Expired	2014-04-14	2015-04-07	2017-05-15	6	CON	NA	1	2016	3
9	1 Albany Park St Mary Street St Andrews KY16 8BP	Licence Expired	2014-04-14	2015-04-07	2017-05-15	6	CON	NA	1	2017	3
10	3 Wallace Street St Andrews Fife KY16 8AN	Licence Expired	2007-01-10	2004-05-28	2007-01-30	3	N/A	NA	0	2004	4
11	3 Wallace Street St Andrews Fife KY16 8AN	Licence Expired	2007-01-10	2004-05-28	2007-01-30	3	N/A	NA	0	2005	4
12	3 Wallace Street St Andrews Fife KY16 8AN	Licence Expired	2007-01-10	2004-05-28	2007-01-30	3	N/A	NA	0	2006	4
13	3 Wallace Street St Andrews Fife KY16 8AN	Licence Expired	2007-01-10	2004-05-28	2007-01-30	3	N/A	NA	0	2007	4
14	3 Ellice Place St Andrews Fife KY16 9HU	Licence Expired	2007-12-10	2004-08-02	2007-03-06	4	N/A	NA	0	2004	4
15	3 Ellice Place St Andrews Fife KY16 9HU	Licence Expired	2007-12-10	2004-08-02	2007-03-06	4	N/A	NA	0	2005	4
16	3 Ellice Place St Andrews Fife KY16 9HU	Licence Expired	2007-12-10	2004-08-02	2007-03-06	4	N/A	NA	0	2006	4
17	3 Ellice Place St Andrews Fife KY16 9HU	Licence Expired	2007-12-10	2004-08-02	2007-03-06	4	N/A	NA	0	2007	4

3 HMO by Years

Figure 5

	year	n_license
1	2004	10
2	2005	19
3	2006	172
4	2007	373
5	2008	629
6	2009	871
7	2010	986
8	2011	1124
9	2012	1079
10	2013	1094
11	2014	1073
12	2015	1193
13	2016	1183
14	2017	1259
15	2018	1262
16	2019	1159
17	2020	1083
18	2021	848
19	2022	700

.

Graph 1

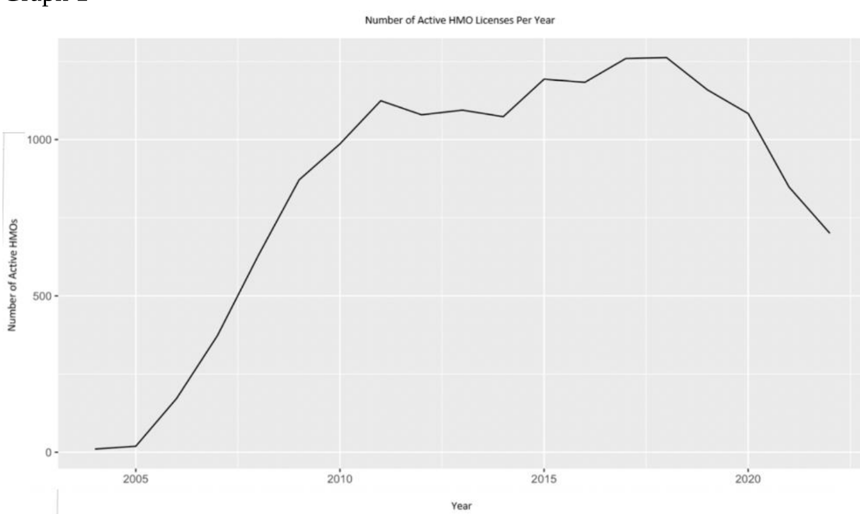
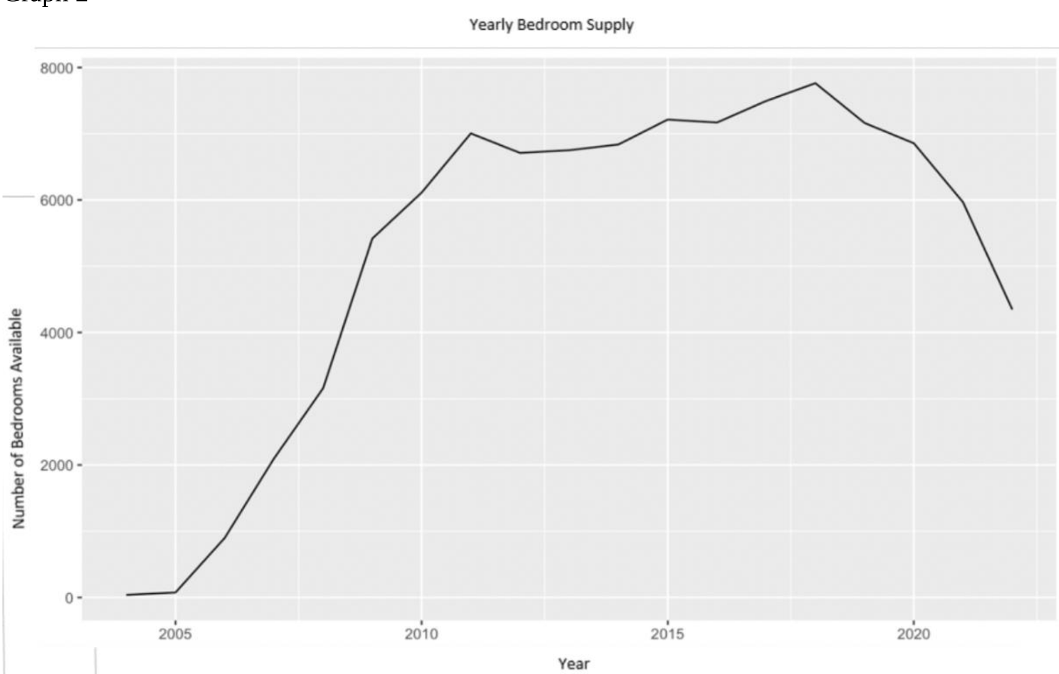


Table 2 and Graph 1 (above) display the total number of active HMO licences every year. As can be seen on the line graph, there is a steady increase up until 2019, where there is a stagnant drop. This is due to the issuance of the Houses in Multiple Occupation in Scotland in 2019. This graph indicates that after the HMO Licence freeze was applied to St Andrews the number of active licences in the following years (2020, 2021 and 2022) decreases. This indicates to us that properties were not being issued or re-issued licences which reflects the policy.

Figure 6

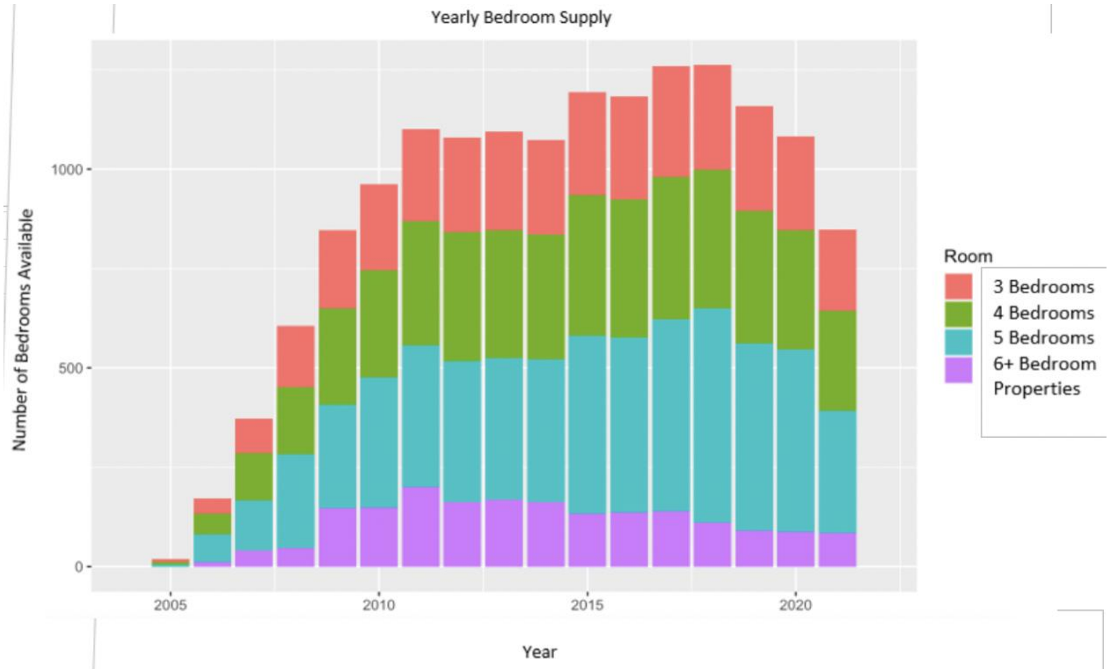
	year	room_3	room_4	room_5	room_6g	total
1	2004	4	4	2	0	38
2	2005	7	7	5	0	74
3	2006	38	53	69	12	898
4	2007	87	119	127	40	2096
5	2008	153	170	235	47	3159
6	2009	197	242	261	147	5419
7	2010	216	270	328	148	6112
8	2011	231	312	357	200	7006
9	2012	238	323	355	163	6711
10	2013	248	321	356	169	6752
11	2014	238	313	360	162	6839
12	2015	258	354	447	134	7215
13	2016	259	347	440	137	7170
14	2017	279	358	482	140	7496
15	2018	263	349	539	111	7764
16	2019	264	334	470	91	7163
17	2020	236	301	459	87	6857
18	2021	205	251	308	84	5966
19	2022	173	211	276	40	4347

Graph 2



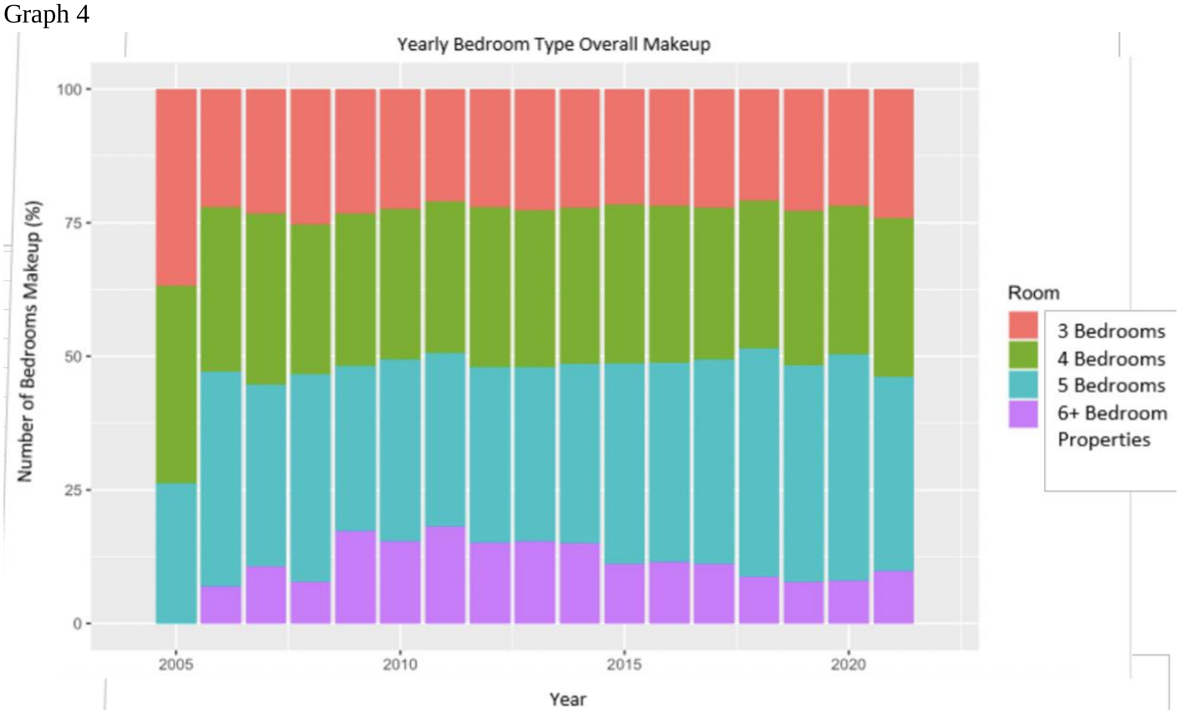
The above table (3) and graph (2) look at the number of available bedrooms available each year. This is useful as the number of bedrooms acts as a supply indication of how many potential people can stay in a HMO licensed house. Once again, there is a steady increase until about 2019, where the number of available bedrooms available to rent decreases drastically, which we have interpreted as a decline in the supply of HMO-licenced properties in the St Andrews rental market as a consequence of the HMO-licence policy cap.

Graph 3



This graph (3) displays the yearly bedroom supply through a bar plot, taking a particular look at the bedroom type makeup. As can be seen, the majority of the makeup of bedrooms in a HMO licence property is a five bedroom. The graph also shows the decrease in HMO-licenced properties after the HMO licence cap policy implemented in 2019.

Graph 4



The above graph (4) is a percentile bar plot looking at the makeup of the type of bedroom per year. Each year is at 100% and displays the breakdown of the type of bedrooms for the year. The proportion of four

bedrooms seems to be seems to stay relatively constant since 2012. We produced this bar plot to study the characteristics of the rental market and if this showed any significant trend pre- and post-HMO licence freeze. The plot shows that there hasn't been a drastic difference in the bedroom size of properties.
Note: Table of percent breakdown can be found in Appendix 3.

Figure 7

	year	CON	GRANT
1	2011	1	273
2	2012	18	334
3	2013	91	199
4	2014	52	168
5	2015	199	353
6	2016	51	219
7	2017	91	229
8	2018	193	210
9	2019	135	167
10	2020	119	52
11	2021	150	15
12	2022	208	40

Graph 5

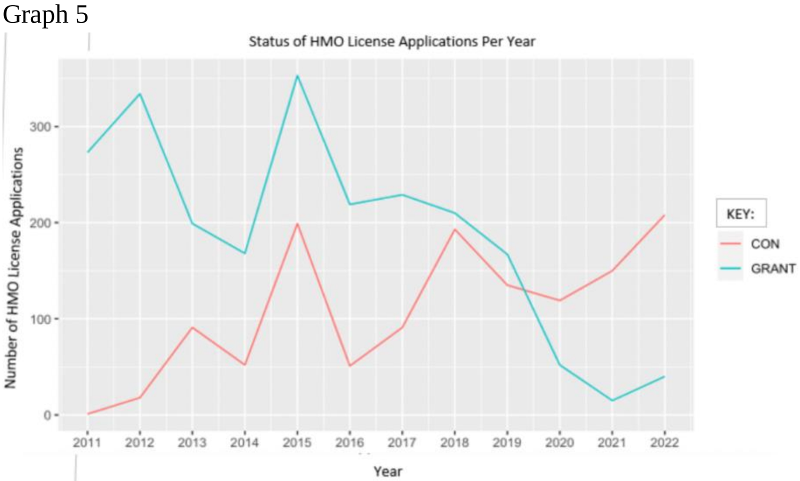


Table 4 and *Graph 5* display the results of every license application since 2011. This table and graph only considers data entries starting from 2011 because of the number of "N/A"s in the years 2004-2010. Fife Council did not store the status of the licences applied at that time frame, so a majority during those years have "N/A". As a result, the above table and line graph depicts the number of "CON" (Rejected) and the number of "GRANT" (Granted). As seen in the line graph above, starting from the year 2019, there is a huge decline in the number of granted HMO licences and an increase in the number of rejected HMO licences.

V. Web Scraping - Data Coding

What are we trying to get from our data?

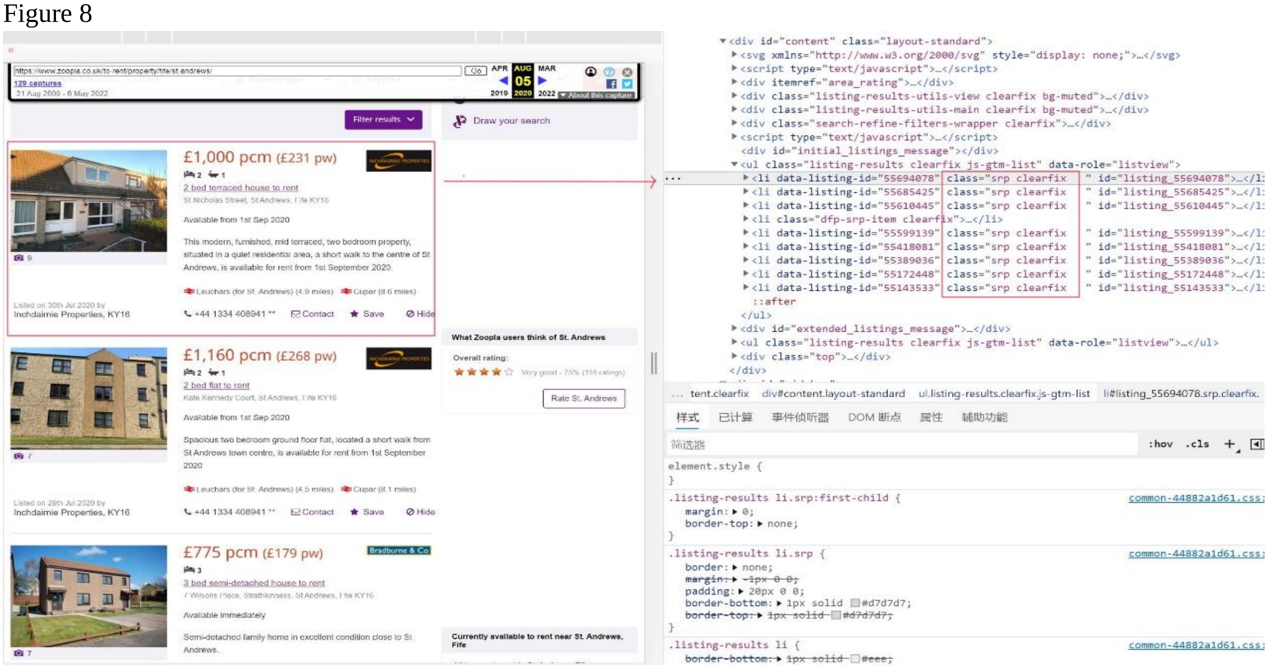
The goal of web scraping the past and current rent prices for properties in St. Andrews is to compare and analyze the evolution of rent prices throughout the past decade and how it was affected due to the HMO freeze. We decided not to find the average annual rental price and calculate the percentage change each year, as in our data, some years have 40 data points while others have 100. Therefore, we didn't want a rental price change to be due to a sampling issue, where one year contained data from a majority of high end properties, while other years had a large concentration of lower end properties. Therefore, at this stage we choose to track and analyze data through histograms for each year (Graph 6), percentage comparisons (Graph 7), as well as taking specific properties where we have rent prices for every year (Graph 8), and evaluate how the rent price has evolved.

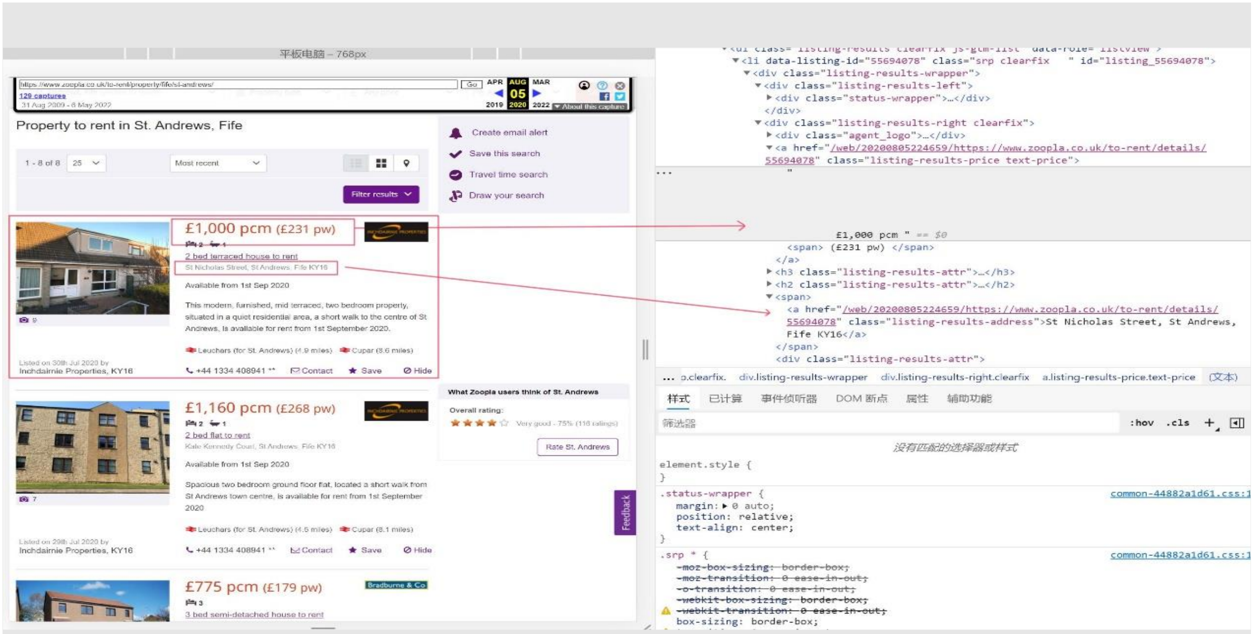
Methodology

Web scraping, as its name suggested, is data scraping used for extracting data from websites. Although this process can be done manually, it is much more efficient to use coding, which enables us to quickly fetch and extract a large amount of data from websites. The full code of web scraping for Zoopla with comments for each line is shown in Appendix 1. The following is an explanation of logics of our web scraping process and some key methods we used.

When doing web scraping, we are actually extracting information from HTML (Hyper Text Markup Language), the code that is used to structure a web page and its content. In order to find information that we are interested in, we need to first observe the structure of HTML and figure out the crucial features that help us locate certain lines of codes. When scraping Zoopla webpage below, for instance, it can be observed that for each square of property information, there is a corresponding section of code in HTML with same class name. We can then use tools from python to fetch all sections of codes with that class name. Within each section, we use similar method find to get specific information like price address.

Figure 8





Several packages and libraries need to be imported to implement web scrapping and further data manipulation in Python. The use of these imported elements saves a lot of time writing tools that have already been written by others. Table one briefly introduces the essential packages we used in the code.

Figure 9

Package & Library	**Description**
Beautiful soup	Beautiful Soup creates a parse tree for parsed pages that can be used to extract data from HTML (web code)[4]
Selenium	Selenium is a powerful tool for controlling web browsers through programs and performing browser automation.
Requests & re	Requests library is one of the integral parts of Python for making HTTP requests to a specified URL
Pandas	Pandas provides a number of tools for data manipulation
Numpy	Numpy provides many mathematical operations
plt	Plt is used for graph plotting
curve_fit	Curve_fit helps find best fits parameters for given data sets.

The code mainly consists of two parts. The first part conducts the web scraping for the Wayback machine to harvest all available past URLs of Zoopla. The second part then conducts another web scrapping for each URL obtained to fetch housing information that we want.

[4] Beautiful Soup (HTML parser), Wikipedia

When doing web scraping of Wayback machine, we separate its URL into three parts, with first and third part being simple string, and second part being a value from a list of years. (Shown in equation) The purpose of this process is to cope with changes of Wayback machine's URL when searching for different years. By observing the pattern of changes over time, we noticed that the orange part of the URL always corresponds to the year of searching. This means that we will be able to open Wayback machine for different years by just modifying this single part of URL. To load and acquire webpage, we use both `driver.get` from `webdriver` in library Selenium and `requests.get` from library Requests. The advantage of using Selenium is that it helps us scrape complicated web pages with dynamic content. By contrast, Requests offers a much faster scraping speed.

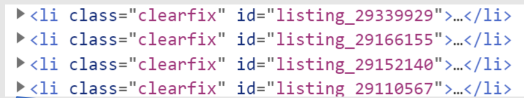
Equation 4

```
#create a list for years we would like to search
years = ["2012", "2013","2014","2015","2016","2017","2018","2019","2020"]

#seperate the url of wayback into different parts
url_Wayback_1 = "http://web.archive.org/"
url_Wayback_2 = "1201000000*/https://www.zoopla.co.uk/to-
rent/property/fife/st-andrews/"

#use for loop that goes through each year
for year in years:
  # Load selenium webdriver with the each year's Wayback url
  driver.get(f"{url_Wayback_1}web/{year}{url_Wayback_2}")


http://web.archive.org/web/20121201000000*/https://www.zoopla.co.uk/to-
rent/property/fife/st-andrews/
```

One of the most important steps in web scraping is to locate and fetch the information from the massive web code. The way we did it is to use tools from Beautiful Soup: `find` and `find_all`. In the simplified code below, `find_all` is first applied to soup (the original web code) with search criteria shown in the bracket. This returns a list of sections of codes matching given conditions. The `find` is then used on every section in the list to get the exact information.

Equation 5

Sections of code for each square of housing information




```
results = soup.find_all("li", class_="clearfix")
for result in results:
    rent = result.find('a', class_='listing-results-price')
```

Line of code in each section providing price info

Data Organization

Through web scrapping we were able to collect roughly 1400 data points that included: property address, date property was listed (including day, month, year), number of bedrooms, total monthly rent, and URL.

Challenges: During the web scraping process, we faced a series of challenges when using the Way back Machine that limited our ability to obtain data points for past rent prices in St. Andrews. When using the way back machine, you are only able to see the past website history of one specific URL. Therefore, when web scraping, we were required to find websites that listed both the property address as well as the monthly rent price on the same page. This limitation, immediately crossed off a number of websites that we were able to web scrape information from, limiting our data set. Another problem that was associated with the single URL issue, was that we were only able to web scrape from the first page of a website. If a website contained 2 pages of relevant data, we were only able to get the information from the first page.

We then conducted a thorough examination to verify each data point's relevance.

Firstly, we deleted all duplicate data points. Because the data was collected from a rental platform, it was common for the properties to be listed several times throughout the year. Our methodology for deleting duplicate data points was that if the data point had the same address, rent price, and was web scraped in the same year, we assumed it was a duplicate data point and we deleted it. We recognized that some of these data points we classified as duplicates may not be actual duplicates. This is because it could be plausible that there are multiple properties with the same rent price and web scraped in the same year if it was located on long streets such as Market Street or Lamond Drive. We are currently working on solving this issue by web scraping more details from each listing, such as the property image URL or the letting agent company. This will allow us to further investigate these suspected duplicates, and if these details are the same for each property, we will be more confident that indeed they are the same property.

To do so we first separate our dataset based on year and applied `drop_duplicates` to each dataset. After all duplicate records from the same year are removed, we combine these cleaned datasets together.

Equation 6
```
d = {}
#seperate df based on year and remove duplicate
for year in years:
  d[year] = df[df['Date'].dt.year == int(year)]
  d[year] = d[year].drop_duplicates(subset=['Rent', 'Type','Address']
, keep='first')

#combine the processed df together
df_all = pd.concat([d['2012'],d['2013'],d['2014'],d['2015'],d['2016']
,d['2017'],d['2018'],d['2019'],d['2020']])
```

Secondly, we omitted any properties that were not in St Andrews. We first identified the main four non-St Andrews towns that were present in the data set: Dundee, Kirkcaldy, Cupar and Boarshill. We then utilized conditional statements on Excel to identify if the property was in one of those four towns:

Equation 7
```
=IF(OR(ISNUMBER(SEARCH("Dundee",E4)),ISNUMBER(SEARCH("Kirkcaldy",E4))),"TRUE")
=IF(OR(ISNUMBER(SEARCH("Cupar",E4)),ISNUMBER(SEARCH("Boarshill",E4))),"TRUE")
```

The first part of Equation 7 would return "TRUE" if the data point's address contained the words Dundee or Kirkcaldy and the second part would do the same if the address contained Cupar or Boarshill.

To double check these properties were indeed not in St Andrews, we used conditional statements to identify if the properties were in St Andrews:

Equation 8
```
=IF(OR(ISNUMBER(SEARCH("St Andrews",E4)),ISNUMBER(SEARCH("St. Andrews",E4))),"TRUE")
=IF(OR(ISNUMBER(SEARCH("St.Andrews",E4)),ISNUMBER(SEARCH("St.andrews",E4))),"TRUE")
```

Equation 8 would indicate "TRUE" if the property address contained St Andrews. The conditional statements also included four different spelling and spacing of St Andrews to ensure Excel correctly identified the properties (see columns "In St Andrews? Test 1" and "In St Andrews? Test 2" in Figure 7).

We then deleted all data points that indicated "TRUE" to being in a non-St Andrews town and indicated "FALSE" to being in St Andrews.

Lastly, we were left with fifty-four data points that were unclear. They indicated "FALSE" for being in St Andrews and "FALSE" to being in any of the four non-St Andrews towns. Therefore, we manually inputted the addresses into Google to confirm it was in St Andrews (See Column "Manual Check In St Andrews?". Three of the fifty-four "unclear" addresses were not in St Andrews and therefore were deleted from the data set. The rest were addresses within St Andrews but did not explicitly list the town name in the address.

Figure 10

	Date	Rent	Type	Address	Rent per person	Link	In St Andrews? Test 1	In St Andrews? Test 2	**In St Andrews? 1st round decision**	In Dundee or Kirkcaldy?	In Cupar or Boarshill?	**Manual Check- In St Andrews?**	**FINAL DECISION: in St Andrews?**
61	23/07/2014	£1,100.00	1	South Street, St Andrews KY16	£1,100.00	http://web.archive.org/web/2014	TRUE	FALSE	TRUE	FALSE	FALSE		TRUE
47	23/07/2014	£725.00	3	Dairy Cottage, St Andrews, Fife, KY16	£241.70	http://web.archive.org/web/2014	TRUE	FALSE	TRUE	FALSE	FALSE		TRUE
48	23/07/2014	£865.00	3	31 Lamberton Place, St Andrews, Fife, KY16	£288.30	http://web.archive.org/web/2014	TRUE	FALSE	TRUE	FALSE	FALSE		TRUE
249	23/07/2014	£2,875.00	5	Windmill Road, St Andrews, Fife KY16	£575.00	http://web.archive.org/web/2014	TRUE	FALSE	TRUE	FALSE	FALSE		TRUE
49	23/07/2014	£1,250.00	5	Cassindonald, By St Andrews, Fife KY16	£250.00	http://web.archive.org/web/2014	TRUE	FALSE	TRUE	FALSE	FALSE		TRUE
50	23/07/2014	£1,300.00	4	Kilrymont Road, St Andrews, Fife KY16	£325.00	http://web.archive.org/web/2014	TRUE	FALSE	TRUE	FALSE	FALSE		TRUE
78	19/10/2014	£600.00	2	Bonfield Park, Strathkinness KY16	£300.00	http://web.archive.org/web/2014	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
52	19/10/2014	£700.00	1	Church Street, St Andrews, Fife KY16	£700.00	http://web.archive.org/web/2014	TRUE	FALSE	TRUE	FALSE	FALSE		TRUE
53	19/10/2014	£780.00	2	Allan Roberson Dri, St Andrews KY16	£390.00	http://web.archive.org/web/2014	TRUE	FALSE	TRUE	FALSE	FALSE		TRUE
54	19/10/2014	£2,150.00	4	Hepburn Gardens, St. Andrews KY16	£537.50	http://web.archive.org/web/2014	TRUE	FALSE	TRUE	FALSE	FALSE		TRUE
270	19/10/2014	£1,000.00	2	James Foulis Court, St. Andrews KY16	£500.00	http://web.archive.org/web/2014	TRUE	FALSE	TRUE	FALSE	FALSE		TRUE
83	31/12/2014	£650.00	2	Bonfield Park, Strathkinness, Fife KY16	£325.00	http://web.archive.org/web/2014	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
55	31/12/2014	£700.00	2	Roundhill Road, St. Andrews KY16	£350.00	http://web.archive.org/web/2014	TRUE	FALSE	TRUE	FALSE	FALSE		TRUE
56	31/12/2014	£5,000.00	4	Priory Gardens, St. Andrews KY16	£1,250.00	http://web.archive.org/web/2014	TRUE	FALSE	TRUE	FALSE	FALSE		TRUE
57	31/12/2014	£1,500.00	2	The Links, St Andrews KY16	£750.00	http://web.archive.org/web/2014	TRUE	FALSE	TRUE	FALSE	FALSE		TRUE

Figure 7 displays the full process of accurately identifying all St Andrews addresses, culminating in the final column “FINAL DECISION: in St Andrews” being “TRUE” for all data points.

After deleting duplications and non-St Andrews properties, we are left with 478 [5]total relevant records.

[5] Rent_2012_2020_Zoopla

VI. Web Scraping - Results

After obtaining 478 relevant data points of St. Andrews properties from 2012 until 2020, we calculated the monthly rent per person for each property. This allowed us to compare rent prices across different size flats, for example, a three-bedroom vs five-bedroom flat.

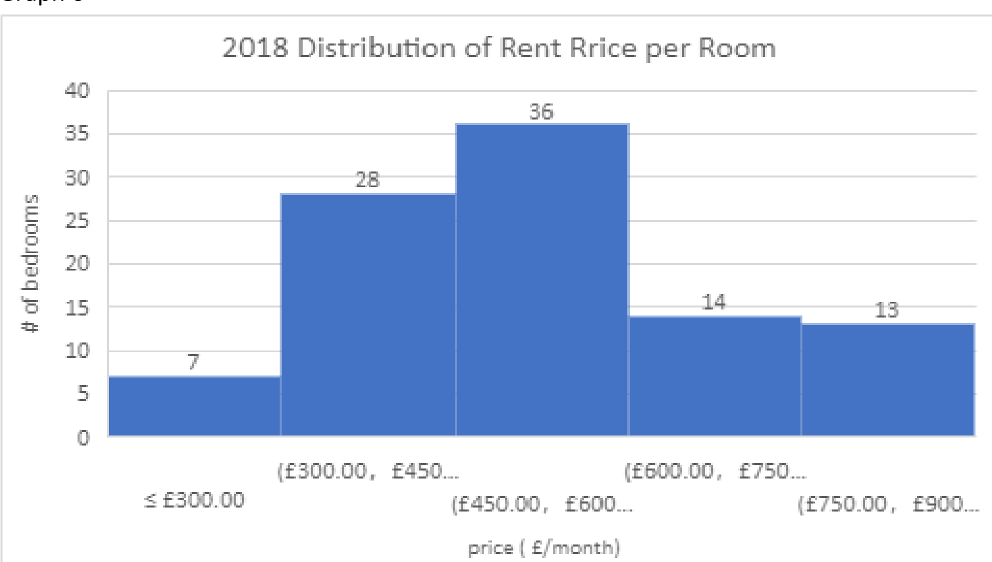
We then plotted the distribution of rent price per room from 2018-2020. We selected these years in hopes of capturing the effect right before (2018) and after (2020) the 2019 HMO freeze. To achieve a more accurate sample, we counted each bedroom as a data point. For example, a 3-bedroom priced at £600/month was counted three times in the data sheet in order to accurately account for the number of rooms available to rent in St Andrews. Figure 8 shows how the four-bedroom property at Fraser Avenue was counted as four individual data points. Please note this duplication of data based on bedroom size, only occurs in order to make graphs 6-8.

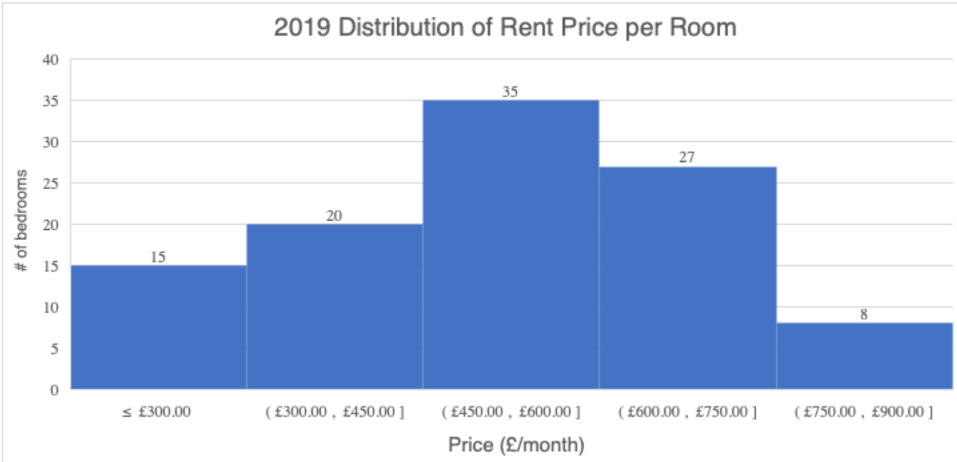
Figure 11

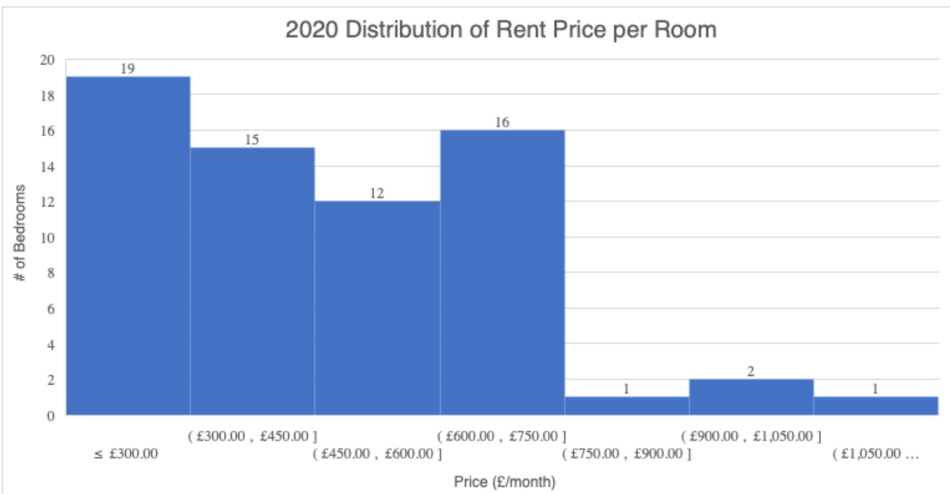
Date	Rent	Type	Address	Rent per pers
06/04/2020	875	4	Fraser Avenue, St. Andrews KY16	£ 218.75
07/04/2020	875	4	Fraser Avenue, St. Andrews KY16	£ 218.75
08/04/2020	875	4	Fraser Avenue, St. Andrews KY16	£ 218.75
09/04/2020	875	4	Fraser Avenue, St. Andrews KY16	£ 218.75
04/06/2020	775	3	7 Wilsons Place, Strathkinness, St Andrew	£ 258.33
04/06/2020	775	3	7 Wilsons Place, Strathkinness, St Andrew	£ 258.33
04/06/2020	775	3	7 Wilsons Place, Strathkinness, St Andrew	£ 258.33

The three parts of Graph 6 show the number of bedrooms within a certain price range. The exhibited results are valuable as the distribution of rent prices of St. Andrews properties that are listed are available to the public to view. We will not be drawing conclusions for the entire rental market in St. Andrews, such as saying the average rent price has increased, as this data set is small in comparison to the entire population. We only collected 98 data points for 2018, 105 for 2019, and 66 for 2020. However, this is the only data that has been available for our group to web scrape. Therefore, without further data, our results will not be more accurate than the ones provided below.

Graph 6

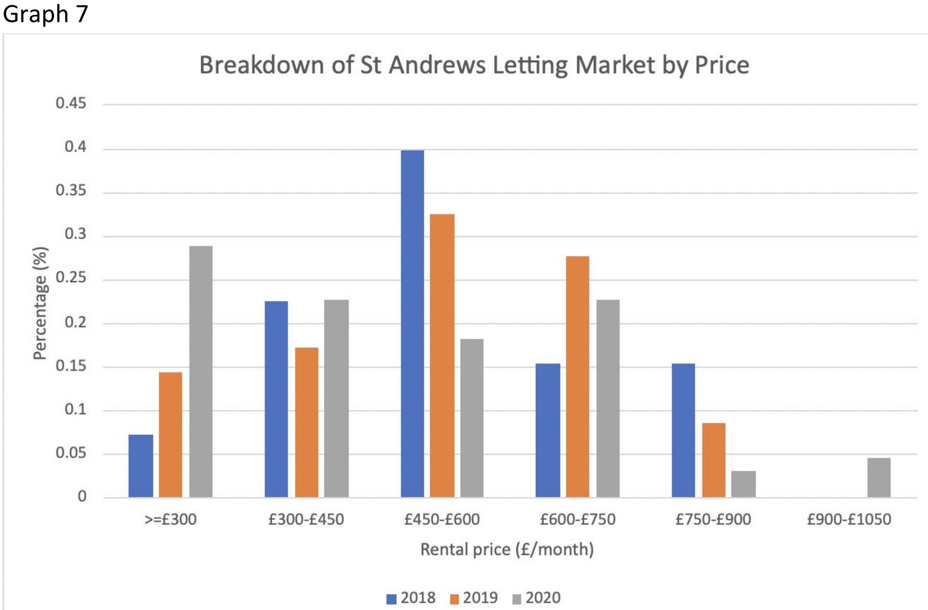






Another limitation is our sample size was different for each year, making a fair comparison difficult.

Graph 7



In Graph 7 we tried to mitigate the problem of different sample sizes by creating a breakdown of the rental market by percentage rather than total number of rooms. In other words, we calculated what percentage of the rooms available fell into a certain price range. For example, in 2018, 40% of the rooms available fell within the £450-£600 price range while in 2020 only 18% of rooms fell within this same price range. However, we need to be careful in interpreting this data. Just because a certain year has a larger portion of higher-priced properties does *not* mean the rental price has increased. This is likely the result of a sampling bias. The sample of rooms is not random as letting sites select which properties to post. Some years may randomly include higher priced properties than other years, making it difficult to identify changing rental price. Also, the data each year is only capturing properties listed by letting agencies. Therefore, each year we fail to capture properties that are privately let.

Another reason we cannot use this data to measure increase in rental price is because in the graphs above we are comparing a mix of different types of properties. Each property has a different quality or size, making the mix of properties each year not comparable. In order to track the increase of rental price we need to compare the price of a single property over several years. We aim to do this is the following paragraph.

Investigation on Ayton House

Realizing that current dataset from Zoopla still have several drawbacks and may produce unreliable analysis result due to differences of these properties on location, durations and availabilities, we try to investigate the change of rent from a specific property to avoid possible bias. Ayton House, a private accommodation at St Andrews with HMO license, is a good choice for analysis. We did some further web scraping to get a more comprehensive dataset of prices for different types of rooms at Ayton House from 2016 to 2022 and created the plot and table shown in Graph 12 and Table 13. Curves displayed in Graph 7 shows lines of best fit for price each type of room, and points around lines represents the actual rents for those rooms. The average rises in rent for different types of rooms as well as corresponding standard deviation (uncertainty) are calculated as shown in the Table below.

TYPE	2 BEDROOM NON-ENSUITE	2 BEDROOM ENSUITE	4 BEDROOM ENSUITE	5 BEDROOM ENSUITE	COMPACT STUDIO	CLASSIC STUDIO	2 LARGE BEDROOM ENSUITE	PREMIUM STUDIO	DIAMOND STUDIO
AVERAGE RISE IN RENT PER YEAR (£)	2.71±1.83	3.11±0.86	5.00 ±1.01	6.11±1.72	6.61±1.64	6.82±2.22	9.57±3.10	10.68±2.12	42
PERCENTAGE INCREASE OF RENT PRICE (%)	1.61±1.08	1.68±0.44	2.73±0.55	3.41±0.96	3.50±0.87	3.53±1.15	5.06±1.63	4.79±0.95	17.00
RENT IN 2022 (£)	197	198	202	200	215	228	216	262	289

Figure 12

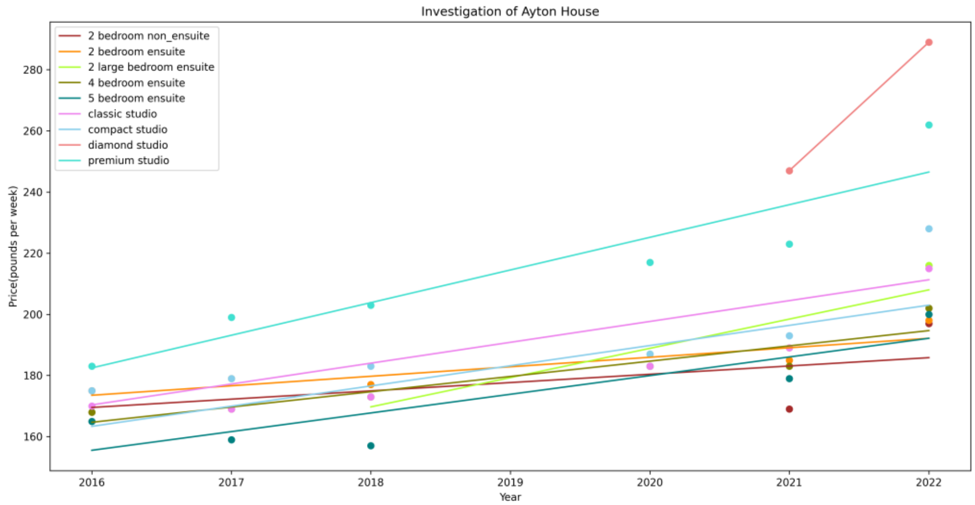


Figure13

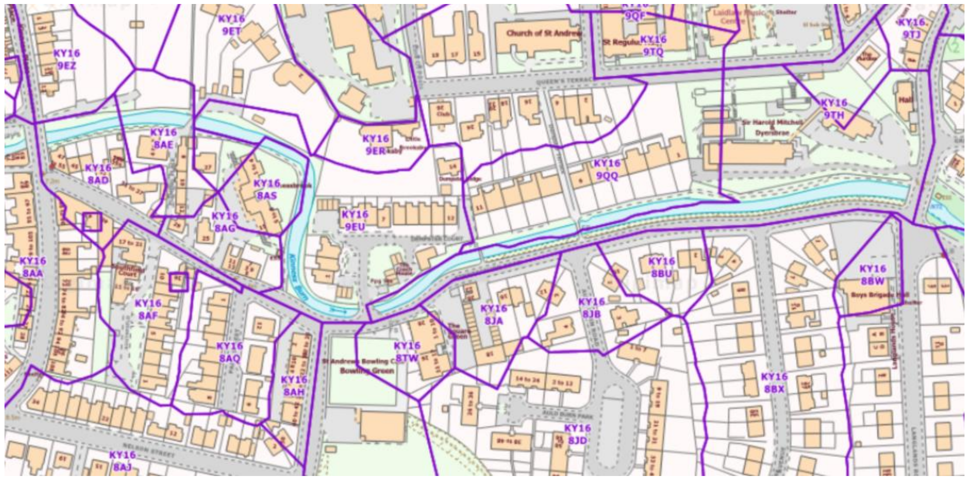
As can be seen from Graph 7 and Table 5, rooms with higher prices generally have both higher increase and percentage increase in rent. The percentage increase of rent for premium studio, for example, is about three times that of 2-bedroom non-ensuite. In order to investigate the effect of HMO freeze, we will find lines of best fit both before and after 2019 to see how expected rises in rent(slopes of best fit line) change. For example, we estimate the price increases by roughly 4.79% for the premium studio each year.

VII. Survey

- **Purpose**
  - The goal of conducting a survey in Spring of 2023 is to gather the most up to date data to understand the full effect of the HMO license cap in St. Andrews on rent prices, total bedrooms occupied, and student's ability to live in St. Andrews as rent prices continue to rise. We hope to gauge students' willingness to pay through their responses, and determine how these vary among postgraduates and undergraduates, dividing students by year of study. We will also consider students' fee status including international, English, or Scottish. Lastly, we will take into account whether students are paying their rent themselves, have a loan for accommodation, or are financially supported by parents/guardians. The data collected from this survey including addresses of properties with HMO licenses, rent, and the number of residents will be integrated into the economic model and will help us develop a map of St. Andrews using ArcGIS displaying HMO properties. Through these representations of the survey data, we will formulate recommended changes to the HMO license cap to Fife Council and the University.

- **Survey Methodology**
  - Target Audience: all St. Andrews Students (including those located in Dundee and those in University accommodation)
    - We are including students living in halls to determine if students choose to live in halls due to the relative discount of halls to some private accomodation options in town. It will also include first year students to determine their willingness to pay as they begin to search for their residence for the following year.
  - Survey Area:
    - The purple line splitting the town into 2 sections along the Kinnesburn is the dividing line, splitting the KY16 postal code into the KY16 8.. and KY16 9.. postal codes. Previous project groups have focused attention on the traditional town conservation area, comprising an area of North Street, Market Street, and South Street. Our team believes that this limited area does not fully capture the student housing market as an increased demand from rising student numbers has pushed many students further out of the conservation area. We looked at using the British National Grid System to identify our target survey area, however we decided that this would not be the most precise way of sorting survey results.

Figure 14

- 

- Students to the north and northwest of the Kinnesburn are in the KY16 9.. postal area and students to the south and southeast of the Kinessburn are in the KY16 8.. postal area.

- It is not possible to capture one screenshot of the survey area of interest showing the individual postalcodes, so here are 4 images showing the boundaries of our area of interest.

Figure 15

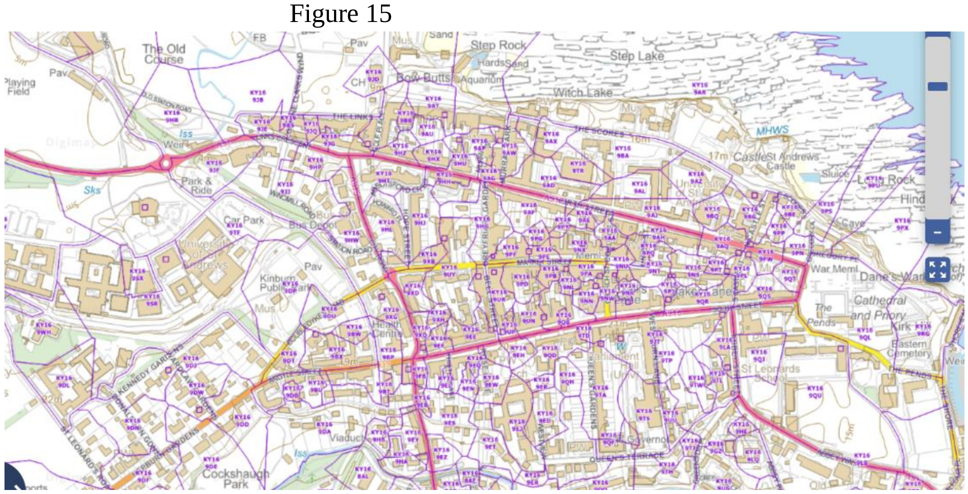
- 

Figure 16

Figure 17

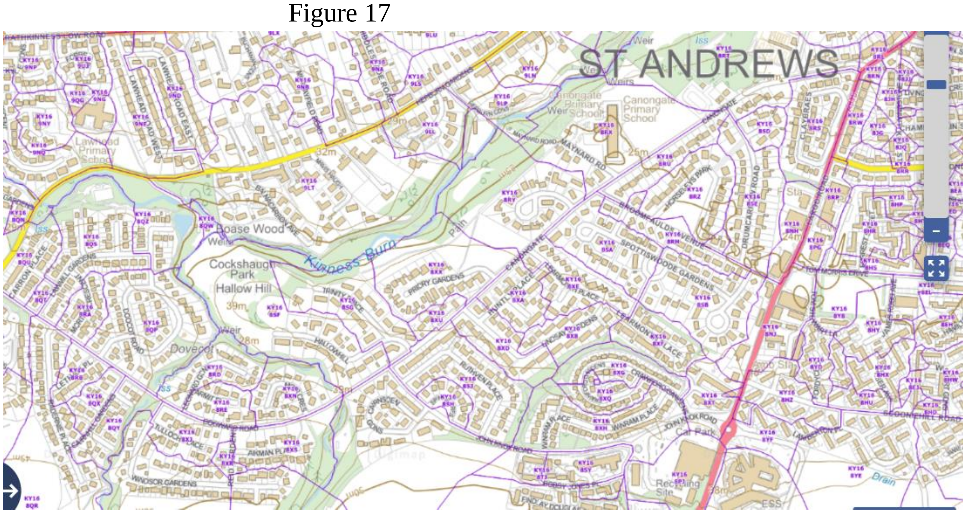
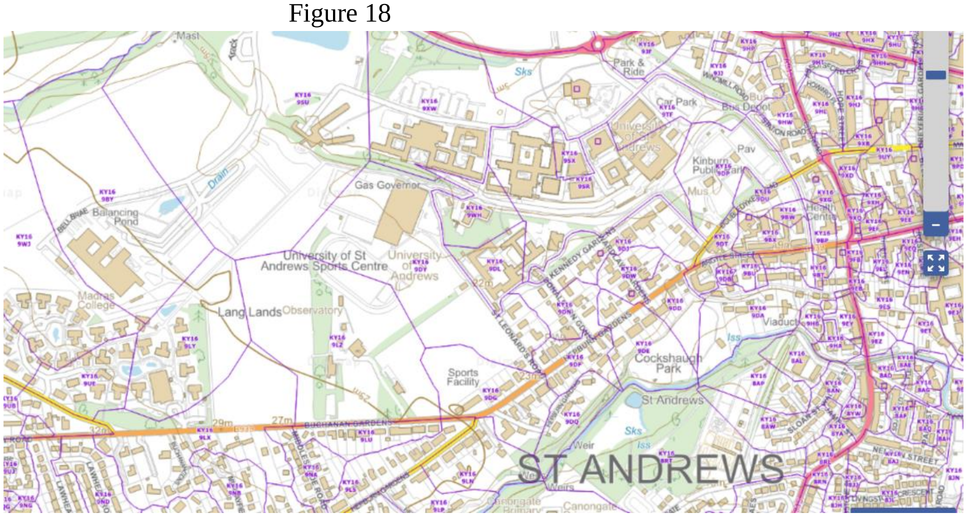


Figure 18



- This postal code map is accessed through Digimaps and the Ordinance Survey mapping overlay.

Survey Delivery and Distribution:

- **Plans for Survey Test Sample:**
  - We plan to conduct a test of our survey before releasing to the whole student body to measure response rates and gather feedback on the survey questions and survey experience.We will email a small sample size of 15-20 students at their university email addresses with a link to complete the survey and the Participant Information Sheet that will be included with the general survey, in compliance with the university research ethics guidelines. However, we will not inform these students that they are a part of a sample group. We will look to see what response rate we have from the sample group and after the test period, email the partcipants to ask about why they chose to respond or not respond and ask for any feedback they may have on the questions. Our largest concern is a lack of responses, which can be detrimental to the usefulness of the data collected and its role within the economic model. This trial run should help us overcome preliminary difficulties and create the most efficient, effective survey structure.

- **Plans for Advertising Survey:**
  - We plan to advertise this survey via Facebook and collaborate with the organization CASH (Committee for Affordable Student Housing).
  - Preserving the anonymity of students is essential to conducting this survey.

- **Plans for single mode of delivery:**
  - The previous HMO Caps in St. Andrews Team Members administered the survey via MySaint and advertised through Facebook. Since our target students are all St. Andrews students, it makes sense to communicate through the university email database as well as class Facebook pages. We will use a Qualtrics survey with one link that can be accessed through both email and Facebook. Additionally, we plan to link the Qualtrics survey to a Facebook post issued to each class to increase responsiveness. We believe sending a mailer will be ineffective, therefore, digital responses are the easiest way to categorise responses for further analysis.

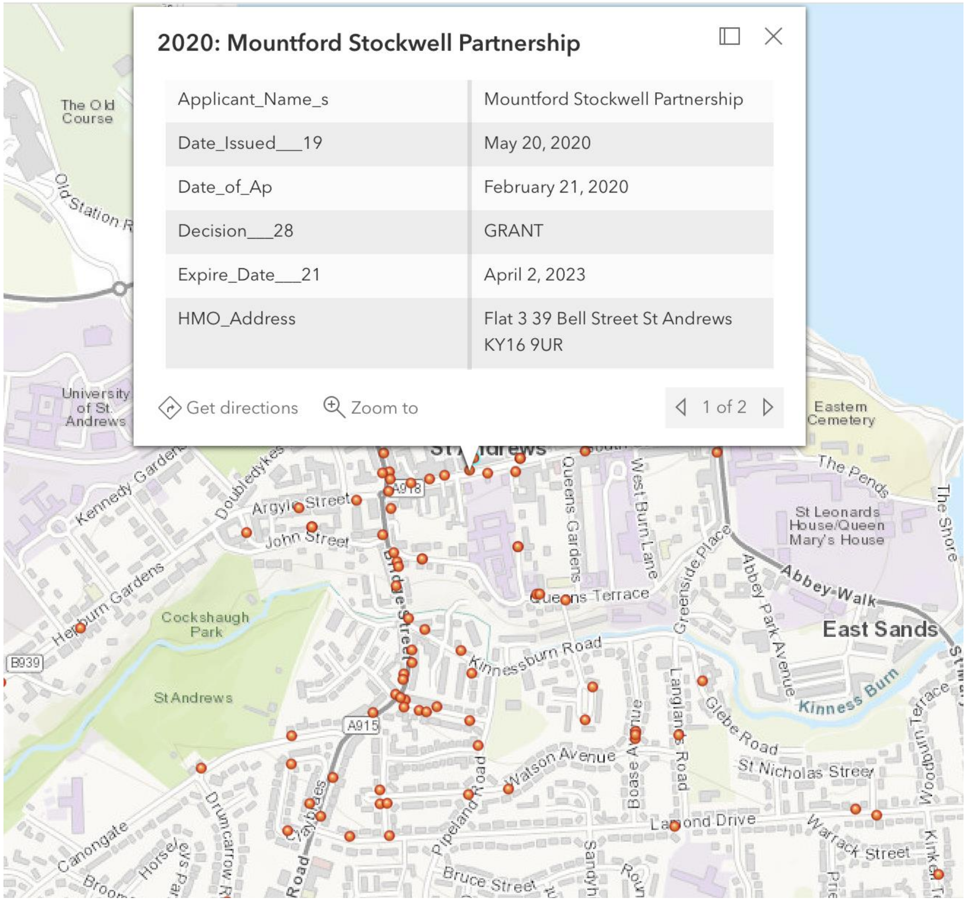
- **A Note on Conflicting Surveys:**
  - We learned in Week 7 of a survey titled ‘St Andrews Private Let Tenant Consultation’ being run by the Fife Council. This survey asks many similar questions to our survey; however, its primary focus seems to be on analysing the effect of the “locked bedroom effect” on the St Andrews housing market.
  - In May of 2022, Fife Council also distributed a survey to a sample of 500 town residents asking about ‘their housing needs and aspirations’. Our survey does not seek data on the town residents, however having this data could be valuable to the project.

        - Our team has emailed the Fife Council HMO licensing office to request they share the results of these two surveys with our team. We have yet to hear back from their office.
    - Campaign for Affordable Student Housing (CASH) previously ran a survey this summer about the student housing crisis after reports of many students still homeless at the beginning of the semester. The university rejected the results of this survey, citing issues with its delivery and methodology.
        - CASH is planning to distribute another survey which may clash with ours. We do not wish to confuse and overwhelm the student body with a multitude of housing surveys, and thus plan to collaborate with CASH on this issue at our next weekly meetings.
        -
- **Survey Data Integration into Economic Model**
    - We hope to uncover an average rent price, the number of HMO properties occupied by students in St. Andrews, and approximately how much students are willing to pay to live in St. Andrews. This information will be calibrated to establish key data points in the economic model, in which case we can estimate the impact on both rent prices and properties available due to the HMO freeze in 2019.
    -
- **Displaying Survey Results:**
    - Our team has been working with secondary data this semester and from web-scraping and other methods, we have been able to extract a large amount of data about HMO licensed and other student properties in town. With our survey hopefully set to drastically expand our data on the housing market, we found it imperative to find a way to visually organize, analyse, and present our data. After researching options, ArcGIS was by far the best option for this task as it allows us to input our excel workbooks as data points on the map and sort the properties using filters such as year of license grant, denied applications, number of rooms, etc.
    - Our team was granted individual licenses to access the software and has spent the last couple of weeks figuring out how to input and sort the data. We have run into the issue of lacking credits needed to store the data that we upload, and we are currently looking for ways around this.
    - Included below is an example of how we will use ArcGIS to visualize our survey results, using data from the HMO licenses granted in 2020.

Figure 19

-
- The figure below provides an example of the information displayed at each data point, showing the applicant name, date of issue, date of approval, expiry date of license, the address of the property, as well as the total number of occupants the license grants (not pictured)

Figure 20



-
- As none of us have used ArcGIS before, we are still figuring out the software. We hope to gain familiarity with the program using our team's secondary data so that visualizing the survey data next semester will be a much smoother process.

VIII. Survey Process

- **Intended Timeline**
  - We plan to launch the survey in the Candlemas Semester (early 2023). To make sure that we are ready for this launch, we have used the Martinmas semester to design the survey and will submit an ethics application to have our survey reviewed by the School of Economics Ethics Committee. Our milestones for staying on track with our timeline are listed as follows:
  - By end of Week 7: finalize list of survey questions
  - By end of Week 8: Create preliminary survey via Qualtrics and decide on what, if any, survey incentives to include
  - By end of Week 9: Submit Ethics Application for review
  - By end of Week 10: Create marketing plan for survey to maximise responses.

  - If survey is approved
    - Begin designing adverts on Facebook, print posters, collaborate with CASH
    - Decide how to conduct the test run of the survey to 10-15 students
  - Semester 2: administer survey to student body
    - Begin by releasing test survey to sample of 15-20 participants
    - Based on results of survey test, modify survey to increase response rate and improve survey experience.
- Official ethics-application-form 2022-23.docx
- Our ethics application, linked above, expands on each of the following sections in much greater detail.
  - Risks and Benefits
    - Students disclose personal data such as where they live, how much they pay in rent each month, where they are from, and if they have moved while studying. Any of these could be potentially triggering based on past personal experiences such as financial struggle, difficult home lives, or a change in circumstance. This data is very sensitive and would be distressing if breached, so data protection is a key priority of our team.
    - The benefits of gaining this information are numerous, as it would provide us with the most accurate and up to date information with which to populate our model.
  - Gaining Voluntary and Informed Consent
    - Participation is optional and names are kept anonymous. However, as the town is very small and the student body is close-knit, publishing the rent at each address could reveal sensitive information. Should we decide that we want to publish this data, it will be made clear to students before they take part in the survey.
  - Confidentiality and Data Management
    - Preserving the anonymity of students is essential to conducting this survey.
    - We will conduct the survey via a secure computer program (Qualtrics) and limit access to the results to strictly the survey team, which will help to ensure confidentiality.

    - Risks to participants such as distress or reputational harm
        - Students may be reminded of stressful situations such as managing personal finances, receiving loans, or struggling to find a place to live.
        - Since every response will be saved under a pseudonym, there is minimal risk to individual data exposure through sharing addresses and rent prices.
- The final steps we need to complete before finishing our ethics application are to decide on the incentives, mockup the survey using Qualtrics, and upload the Participant Information Sheet and Consent Sheet to the application's appendix document checklist. Find out if our survey delivery mode requires the Participant Debrief form to be sent to participants after survey.

IX. Survey Structure

Question	*Why?*	*Type of Response?*	*Potential Issues with It?*	
By how much would your rent have to increase by for you to consider leaving St. Andrews?	WTP *could also use dollar amounts to represent increase because of model calibration	25-50 pounds 50-75 pounds 75-100 Pounds 100-200 Pounds More than 200 pounds	Responses are likely determined by one's personal or family income, which can be sensitive and a source of emotional distress. We are not explicitly asking about one's income levels for this reason, but feel this question is essential in estimating students' willingness to pay.	
Does your property have an HMO license (yes/no)	This data will help us plot HMO properties on a map of St. Andrews using ArcGIS.	yes/no	We will need to define the term 'HMO' because some students may not be aware of what this is. Furthermore, we should elaborate on what it means to not have an HMO license. Just because the responder may be a student who lets a property from an agency or a private landlord, this does not mean their property is licensed.	
Have you lived in your property for more than one year? If so, has your rent increased?	This provides insight on annual changes in rent prices and/or if students have moved accommodation based on the cost of living	yes/no, with a fill in the blank if yes is selected	This asks students to disclose their current living status, which may be quite sensitive due to the stresses of paying rent, finding a place to live, and sharing a space with others. If an individual moved out of a property before their contract ended, this may allude to the discomfort of rising prices and/or a change in financial circumstance.	
What is your primary means of transit to town each day?	Do students walk, bike, drive themselves, or take public transport based on where they live	drop down menu	This may not be completely representative of how far individuals live from town. For instance, commuting to class on a bike does not necessarily indicate that the responder lives far from town.	
How long would it take you to walk to Tesco?	Measures students' ability to reach the 'center' of town	5-10 minutes, 10-15 minutes, 15-20 mins, more than 20 minutes	Some students may not shop at Tesco, and therefore don't have an estimate for this location. Also, the responses may vary based on personal habits, for instance, walking pace.	

Do you live in halls?	Measures how many students live in University accommodation.	yes/no	This may make the respondents feel their privacy is at risk, especially since this question is needed for background data and is not essential to the purpose of the survey.	
Do you live in university managed property?	Measures how many students live in University accommodation.	yes/no	This may make the respondents feel their privacy is at risk, especially since this question is needed for background data and is not essential to the purpose of the survey.	
How many flat mates do you have, if any?	Through this we can measure the average number of students that share a flat	type in a number	This may feel invasive but is essential for the plotting of HMO properties and estimating the number of bedrooms available for student letting in St. Andrews.	
What is your address?	This data will help us plot HMO properties on a map of St. Andrews using ArcGIS.	type in an address	This may appear as a violation of privacy, threatening the anonymity of students.	
Is your property rented through a letting agency? private landlord? Do you own your accommodation?	Measures how many students do not live in University accommodation.	select one	Risks the anonymity of respondents and may feel invasive of one's privacy	
What is your fee status? International, Scottish, RUK (Rest of UK), EU	Affects their ability to pay	select one	Risks the anonymity of respondents	
Are you solely responsible for paying your rent? Do you have an accommodation grant? Do your student loans go towards your rent?	This may affect how much students will pay for certain properties based on their variations of accommodation funding.	yes/no with drop-down menu	This is a highly sensitive subject and may be uncomfortable for some to respond to. Personal or family finances may be a source of stress or a point contention in the lives of students, thus triggering a negative emotional response.	
What is your rent?	This data will be added to the map of St. Andrews assessing how expensive certain areas are.	type in a number	This may appear as a violation of privacy, threatening the anonymity of students.	

Are you an undergraduate or post graduate student?	Used for organization of responses.	select one	Risks the anonymity of respondents	
What year are you?	Used for organization of responses.	1, 2, 3, or 4	Risks the anonymity of respondents	

X. General Lessons

In this section we identify general lessons we have learned regarding workflow, making progress, effective work methods, etc.

Conference: A key takeaway from the VIP conference was that the lack of a returning member to the project had hindered the speed at which the group understood previous years' work. As a result, we found ourselves delayed in determining the direction in which we wanted to focus on for the coming year.

Workflow: Initially we had trouble grasping the main objective of this project as it was very open-ended and lacked a guiding research question. Continuity also posed a major challenge, as we did not know how related our work had to be to the previous year. We learned that splitting into subgroups with a specific individual goals helped mitigate this problem. By clearly identifying a team goal, we were able to lay out more tangible tasks with clear data goals.

Making Progress: We learned the most helpful tool in making progress was sharing a document that tracked our weekly contributions, challenges, and goals for next week. Also, having multiple sub-group work sessions per week, with a quick whole team recap meeting, greatly helped our ability to collaborate.

Effective Methods: Aforementioned, the team had trouble figuring out what needed to be done, allocating work and communicating with the head researcher. The first two weeks of the VIP, there was no collaboration or communication of what was to be accomplished within the team. However, once we split the team into "sub-groups", each having its own group lead created a successful way to streamline communication. A weekly meeting was initiated to review the "To Do" list. The updates were shared with the heads through a shared document so they could advise where needed. By creating a structure for the team, we were able to better collaborate, create timelines for efficiency, and overall, accomplish more.

XI. Future Work

Further work for HMO Register

We have started to look at the attrition rate of the properties and if we can explain this rate by certain characteristics of these properties.

We identified that duplicated rows within the dataset could be also seen as addresses where more than one HMO licence has been issued between 2004-2022.

The following code shows the process of splitting the data frame into two datasets, one containing duplicate entries and the other without.

Equation 9

```
#Looking at dataset that contains duplicated addresses.
#Considering duplicates those that remain in our dataset as properties
where they have had their licence renewed
data_duplicate <- data[which(duplicated(data$Address)==T),]
#Dropout includes properties which have only had a HMO licence issued
once across the whole time period
data_dropout <- data[which(duplicated(data$Address)==F),]
```

We also saw that we could check this on Excel by assigning a value of 1 to all rows that have a duplicate in the sheet, and a 0 to the rows that do not.

Equation 10

```
=--(COUNTIFS($A$2:$A$4919,$A3)>1)
```

When totaling the number of entries in the `data_duplicate` subset and comparing to the rows in Excel assigned the value 1, we noticed there was a significant difference between the results of the two methods.

When manually inspecting the R data set, we noticed that some there were duplicate properties in the `data_dropout` set which should have been included in the `data_duplicate` set. Upon inspecting the Excel results, we also noticed that due to the difference in address spelling some properties were not counted as duplicates when they should have been.

Therefore, an important focus of the rest of the semester will be to develop code to determine the attrition of HMO properties in our database.

Relating web scrapping to the project goal

Due to substantial obstacles encountered (e.g. sampling bias, origin of data, etc.) we have been unable to accurately link the web scrapping data prodcued to the overall project goal. We then decided to conduct a case study on Ayton House to provide more information on how the scrapping could be linked. Another avenue we are yet to explore combining both databases and dicuss the above approaches more below.

Link Ayton House data to HMO freeze

For price analysis of Ayton House case, firstly we will figure out if linear model is the best function to fit for change of rent price. Once a suitable is chosen, we will fit data separately for prices before and after HMO freeze to analysis its effects. Taking the impact of inflation rate & exchange rate into consideration, this result should enable us to conduct a more accurate analysis.

Combining web scraping and HMO database

In the coming weeks we aim to combine our two data sets, the tidied HMO register and the web scraped rental database. The primary way we wish to combine these is linking rent information to addresses in the HMO database. If we find properties where we have both HMO licence information and rental price changes we can use these to determine the supply, demand and consequently willingness-to-pay of the individual(s) renting this property.

Geographic Mapping
After combining the data sets of the HMO database and web scraping, we hope to geographically map each active licenced house in a given year, possibly sorted by bedroom number. With this, there will be a visual indicator of the saturation of HMO licence carrying housing in St Andrews that will allow us to conduct analysis on the effect of the geographic location of a HMO licenced house before and after the 2019 HMO freeze.

Further Work for Survey Team
Our next steps will be to draft our survey with the questions from our question bank and to decide on what, if any, incentives we will include to increase participation rates. Once these two final pieces are completed, we will be ready to submit our ethics application for review by our supervisors and then upon their approval submit to the School of Econ Ethics Committee for their review

References

"Beautiful Soup (HTML Parser)." *Wikipedia*, 17 Jan. 2022, en.wikipedia.org/wiki/Beautiful_Soup_(HTML_parser).

Appendix 1 – Data Modelling

```
#Installing packages and libraries we will need to use.
install.packages("tidyverse")
library(tidyverse)
library(readxl)
install.packages("dplyr")
library(dplyr)
library(sampling)

#Reading the Excel file into R
HMO_statistics_modified_version <-
read_excel("/Users/aoifedoyle/Downloads/HMO-Public-Register-Modified-
Version.xlsx")
#Removing empty columns
df <- HMO_statistics_modified_version %>% select(-2, -3, -4, -5, -9, -10, -
11, -13, -14, -15, -17, -18, -20, -22, -23, -24, -26, -27, -29, -30, -31, -
32:-56)
#Renaming the remaining columns for ease.
colnames(df) <-
c('Number','Name','Ward','Address','Status','Applied','Issued','Expiry','Occu
pants','Decision')
#Removing any data points that were blank in the Name column as this is
repeated data on the same house.
df <- df[!is.na(df$Name),]
#Removing any HMO entries from other postal code areas in Fife.
df <- df[!grepl('KY1 |KY2 |KY3 |KY4 |KY6 |KY7 |KY8 |KY10 |KY11 |KY12 |KY14
|KY15 |DD6 |FK10 |KY16 0UG|KY16 0FE|KY16 0HF|KY16 9SQ|Hotel | Prior Muir |
prior Muir| Dunfermline|Kirkcaldy|Cupar', df$Address),]

#After manual checking of the dataset the following code was to identify and
remove any entries that were not caught by the above code.
#Eliminate any empty rows that were created from converting the PDF to Excel
and into R.
df <- df[!is.na(df$Occupants),]
#Remove data from wards that are not in the area of focus
df <- df[!grepl('NW13|NW18', df$Ward),]
#Accounting for the HMO licences issued for hotels in St Andrews. Making the
assumption that these hotels would not be considered rental properties for
students.
df <- df[!grepl('Hotel', df$Name),]
```

```
#There are duplicate entries of properties where multiple people have their
name on the application form and this has created more than 1 row for that
property.
#The easiest way to remove these is in Excel using conditional formatting so
we exported the dataset to Excel to sort this.
df
install.packages("writexl")
library(writexl)
write_xlsx(df,"/Users//aoifedoyle//desktop//Dataset.xlsx")

#Once the dataset was cleaned up in Excel we imported it back into R to
continue with data analysis
library(readxl)
data <- read_excel("Desktop/Full_HMO.xlsx")
#made an extra column for year
data$year <- as.POSIXlt(data$Issued)$year +1900
#creates each year into data frames and stores in IssuedYear
IssuedYear <- list()
#Creating a loop
for ( i in 2004:2022){
  dat<- as.data.frame(data[data$year == i,])
  IssuedYear <- append(IssuedYear,list(dat))
}
#Call by
IssuedYear[1] #1 = 2004, 2 = 2005, ..., 19 = 2022
#Can save each year as a separate dataframe using the following code.
#Saves all licences issued in 2004 into a dataframe.
IssuedYear2004 <- as.data.frame(IssuedYear[1])  1 = 2004, 2 = 2005, ..., 19 =
2022

#The following code can be used to filter the dataset by year and number of
occupants (number of potential bedrooms in the property).
library(dplyr)
data %>%
  filter(data$year==2004, data$Occupants==3)

#Created a script for searching for the number of active licences during a
time period.
data$Applied = as.Date(data$Applied, "%Y-%m-%d")
data$Issued = as.Date(data$Issued, "%Y-%m-%d")
data$Expiry = as.Date(data$Expiry, "%Y-%m-%d")
data_subset <- subset(data, Issued>= "2017-01-01" & Issued<= "2017-02-01")

#Looking at dataset that contains duplicated addresses.
#Considering duplicates those that remain in our dataset as properties where
the HMO licence has been reapplied for and renewed.
data_duplicate <- data[which(duplicated(data$Address)==T),]
```

```
#Dropout includes properties which have only had a HMO licence issued once
across the whole time period
data_dropout <- data[which(duplicated(data$Address)==F),]
```

Appendix 2 – Implementing HMO Database to Create Models

```
library(readxl)

hmo <- read_excel("Downloads/HMO By Years (1).xlsx", sheet = "Total HMO",
n_max = 6000)
View(hmo)
str(hmo)

install.packages("dplyr")
library(dplyr)
install.packages("lubridate")
library(lubridate)
install.packages("tidyverse")
library(tidyverse)

#number of HMOs that have starting issue date for each year
hmo$year <- year(hmo$`Date Issued`)
table(hmo$year)

#Add "filler" end date to most recent data
hmo[is.na(hmo$`Expire Date`), ]$`Expire Date` <- as.POSIXct("2022-12-30", tz
= "UTC")
hmo$yearrange <- year(hmo$`Expire Date`) - year(hmo$`Date Issued`) + 1

try <- hmo[315,]

#Finds number of active licenses per year
rows <- c()
years <- c()
for (i in 1:nrow(hmo)){
  rows <- append(rows, rep(i, hmo$yearrange[i]))
  newyear <- hmo$year[i] + 0:(hmo$yearrange[i]-1)
  years <- append(years, newyear)
}
hmo2 <- hmo[rows, ]
hmo2$year <- years

#Prints table and graph of annual total active licenses each year from 2004-
2022
count_license <- hmo2 %>% group_by(year) %>% summarise(n_license = n())
count_license %>%
  ggplot() +
  geom_line(aes(x = year, y = n_license)) +
```

```
  xlim(2004, 2022)

#Counts and prints table of total number of available bedrooms from 2004-2022
count_rooms <- hmo2 %>% group_by(year) %>%
  summarise(room_3 = sum(`Tot Occs` == 3, na.rm=T),
            room_4 = sum(`Tot Occs` == 4, na.rm=T),
            room_5 = sum(`Tot Occs` == 5, na.rm=T),
            room_6g = sum(`Tot Occs` >= 6, na.rm=T),
            total = sum(`Tot Occs`, na.rm = T))

#barplot graph annual total number of available bedrooms from 2004-2022
bar_count <-
count_rooms %>%
  pivot_longer(room_3:room_6g, names_to = "Room", values_to = "Count")
ggplot(bar_count) +
  geom_bar(aes(x = year, y = Count, fill = Room), stat = "identity")+
  xlim(2004, 2022)

#barplot percentage graph of the makeup of type of available rooms from 2004-
2022
bar_percent <-bar_count %>%
  group_by(year) %>%
  summarise(Room = Room, Percent = Count/sum(Count) * 100) %>%
  ungroup(year)
ggplot(bar_percent) +
  geom_bar(aes(x = year, y = Percent, fill = Room), stat = "identity")+
  xlim(2004, 2022)

#Line graph of annual total number of available bedrooms from 2004-2022
count_rooms %>%
  ggplot() +
  geom_line(aes(x = year, y = total)) +
  xlim(2004, 2022)

#Counts and prints table of decision of each HMO license from 2011-2022
hmo_2011 <- hmo %>% filter(year(`Date Issued`) >= 2011)
table(hmo_2011$Decision)

license <- hmo_2011 %>%
  group_by(year) %>%
  summarise(CON = sum(Decision == "CON", na.rm = T), GRANT = sum(Decision ==
"GRANT", na.rm = T))

#Line graph of number of approved/denied decisions of each HMO license from
2011-2022
ggplot(license) +
  geom_line(aes(x = year, y = CON, color = "CON"))+
  geom_line(aes(x = year, y = GRANT, color = "GRANT")) +
  scale_x_continuous(breaks = c(2011:2022)) +
```

```
  labs(y = "Count", x = "Year")
```

Appendix 3 – Percentile Bar Plot Table

	year	Room	Percent
1	2004	room_3	40.000000
2	2004	room_4	40.000000
3	2004	room_5	20.000000
4	2004	room_6g	0.000000
5	2005	room_3	36.842105
6	2005	room_4	36.842105
7	2005	room_5	26.315789
8	2005	room_6g	0.000000
9	2006	room_3	22.093023
10	2006	room_4	30.813953
11	2006	room_5	40.116279
12	2006	room_6g	6.976744
13	2007	room_3	23.324397
14	2007	room_4	31.903485
15	2007	room_5	34.048257
16	2007	room_6g	10.723861
17	2008	room_3	25.289256
18	2008	room_4	28.099174
19	2008	room_5	38.842975
20	2008	room_6g	7.768595
21	2009	room_3	23.258560
22	2009	room_4	28.571429
23	2009	room_5	30.814640
24	2009	room_6g	17.355372
25	2010	room_3	22.453222
26	2010	room_4	28.066528
27	2010	room_5	34.095634
28	2010	room_6g	15.384615
29	2011	room_3	21.000000

Appendix 4 – Code for web scraping

```
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
import pandas as pd
import time
import requests
import re

#set header
headers = {
```

```
  "User-Agent":
  "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/86.0.4240.111 Safari/537.36 Edg/86.0.622.51"
}

#selenium set up
chrome_options = Options()
chrome_options.add_argument('--no-sandbox')
chrome_options.add_argument('--disable-dev-shm-usage')
driver = webdriver.Chrome(options=chrome_options)

#create a list for years we would like to search
years = ["2012", "2013","2014","2015","2016","2017","2018","2019","2020"]

#seperate the url of wayback into different parts
url_Wayback_1 = "http://web.archive.org/"
url_Wayback_2 = "1201000000*/https://www.zoopla.co.uk/to-
rent/property/fife/st-andrews/"

#create a list for link
link_list = []

#use for loop that goes through each year
for year in years:
  # Load selenium webdriver with the each year’s Wayback url
  driver.get(f"{url_Wayback_1}web/{year}{url_Wayback_2}")

  # Give some time for the browser to load the content
  time.sleep(3)

  # Use beautiful soup to get the web page
  soup = BeautifulSoup(driver.page_source, 'lxml')

  # limit the web page to only class "calendar-grid"
  results = soup.find("div", class_="calendar-grid"

  #find urls in results and append it into link_list
  for link in results.find_all("a", href=True):
    link_list.append(link['href'])

#Create a list for dictionary
Housing_list = []

#go through links in link list from previous part
for link in link_list:
  link = "http://web.archive.org" + link
```

```
#use requests to load the page
r = requests.get(link, headers=headers)

#use beautiful soup to get the HTML
soup = BeautifulSoup(r.text, "html.parser")

#find all sections of web code with class "clearfix"
results = soup.find_all("li", class_="clearfix")

#fetch date from urls
date = link[27:35]

# go through each section of web code with class "clearfix"
for result in results:

  # find rent, address, number of rooms from this section
  Rent = result.find('a', class_='listing-results-price')
  Address = result.find('a', class_='listing-results-address')
  Type = result.find('h2', class_='listing-results-attr')

  #use if condtion to eliminate situtaions where data can't be obtained
  if Rent != None and Address != None and Type != None:

    #modified format of obtained information
    Rent = Rent.text.replace(',', '')
    Rent_list = re.findall(r'\b\d+\b', Rent)
    Type  = ''.join(filter(str.isdigit, Type.text))

    #add modified information into a dictionary
    if Type != '':
      Type = Type[0]
      if date == "20120109":
        Housing = {
          'Date': date,
          'Rent': int(Rent_list[1]),
          'Type': Type,
          'Address': Address.text,
          'Link': link,
        }
      else:
        Housing = {
          'Date': date,
          'Rent': int(Rent_list[0]),
          'Type': Type,
```

```
            'Address': Address.text,
            'Link': link,
          }

    #add obtained dictionary into the Housing_list
    Housing_list.append(Housing)

driver.quit()
#change the list of dictionary to dataframe
df = pd.DataFrame(Housing_list)
#change Date from string to datetime
df['Date'] = pd.to_datetime(df['Date'], format='%Y%m%d')
#delete duplicate
df = df.drop_duplicates()
#change the format to csv
df.to_csv('housinginfo.csv')
```

Appendix 5 – Plot & Analysis of Ayton House Price

```
#import packages and libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import curve_fit
from google.colab import files

#read our collected Ayton house data
df = pd.read_csv('Ayton (1).csv')

#change Date info from string to datetime format
df["Date"] = pd.to_datetime(df["Date"])

#extract year of data into a new column
df['year'] = pd.DatetimeIndex(df['Date']).year

#extract the price and year of each type of rooms in Ayton house into numpy a
rray
non_ensuite_2 = df.loc[df['Type'] == '2 Bedroom Apartment Non En-
suite','Rent_week'].to_numpy()
non_ensuite_2t = df.loc[df['Type'] == '2 Bedroom Apartment Non En-
suite','year'].tolist()

ensuite_2 = df.loc[df['Type'] == '2 Bedroom Apartment with En-
suite','Rent_week'].to_numpy()
```

```
ensuite_2t = df.loc[df['Type'] == '2 Bedroom Apartment with En-
suite','year'].tolist()

ensuitelarge_2 = df.loc[df['Type'] == '2 Bedroom Large Apartment Ensuite','Re
nt_week'].to_numpy()
ensuitelarge_2t = df.loc[df['Type'] == '2 Bedroom Large Apartment Ensuite','y
ear'].tolist()

ensuite_4 = df.loc[df['Type'] == '4 Bedroom Apartment with En-
suite','Rent_week'].to_numpy()
ensuite_4t = df.loc[df['Type'] == '4 Bedroom Apartment with En-
suite','year'].tolist()

ensuite_5 = df.loc[df['Type'] == '5 Bedroom Apartment with En-
suite','Rent_week'].to_numpy()
ensuite_5t = df.loc[df['Type'] == '5 Bedroom Apartment with En-
suite','year'].tolist()

classic_studio = df.loc[df['Type'] == 'Classic Studio','Rent_week'].to_numpy(
)
classic_studiot = df.loc[df['Type'] == 'Classic Studio','year'].tolist()

compact_studio = df.loc[df['Type'] == 'Compact Studio','Rent_week'].to_numpy(
)
compact_studiot = df.loc[df['Type'] == 'Compact Studio','year'].tolist()

diamond_studio = df.loc[df['Type'] == 'Diamond Studio','Rent_week'].to_numpy(
)
diamond_studiot = df.loc[df['Type'] == 'Diamond Studio','year'].tolist()

premium_studio = df.loc[df['Type'] == 'Premium Studio','Rent_week'].to_numpy(
)
premium_studiot = df.loc[df['Type'] == 'Premium Studio','year'].tolist()

#set up the plot
_,ax = plt.subplots(1,1,figsize = (16,8))
ax.set_xlabel("Year")
ax.set_ylabel("Price(pounds per week)")
ax.set_title("Individual Investigation of Ayton House")

#scatter the data point of room prices
ax.scatter(non_ensuite_2t, non_ensuite_2, color = 'brown')
ax.scatter(ensuite_2t, ensuite_2, color = 'darkorange')
ax.scatter(ensuitelarge_2t, ensuitelarge_2, color = 'greenyellow')
ax.scatter(ensuite_4t, ensuite_4, color = 'olive')
```

```
ax.scatter(ensuite_5t, ensuite_5, color = 'teal')
ax.scatter(compact_studiot, compact_studio, color = 'violet')
ax.scatter(classic_studiot, classic_studio, color =  'skyblue')
ax.scatter(premium_studiot, premium_studio, color = 'turquoise')
ax.scatter(diamond_studiot, diamond_studio, color = 'lightcoral')

#define th function for best fit(linear at this stage)
def linear_fit(year, k, b):
    return k * year + b

#calculate the fit coefficient and its standard deviation for each type of ro
om at Ayton House
[k1,b1], est_covs1 = curve_fit(linear_fit, non_ensuite_2t, non_ensuite_2)
est_std_devs1 = np.sqrt(np.diag(est_covs1))
[k2,b2], est_covs2 = curve_fit(linear_fit, ensuite_2t, ensuite_2)
est_std_devs2 = np.sqrt(np.diag(est_covs2))
[k3,b3], est_covs3 = curve_fit(linear_fit, ensuitelarge_2t, ensuitelarge_2)
est_std_devs3 = np.sqrt(np.diag(est_covs3))
[k4,b4], est_covs4 = curve_fit(linear_fit, ensuite_4t, ensuite_4)
est_std_devs4 = np.sqrt(np.diag(est_covs4))
[k5,b5], est_covs5 = curve_fit(linear_fit, ensuite_5t, ensuite_5)
est_std_devs5 = np.sqrt(np.diag(est_covs5))
[k6,b6], est_covs6 = curve_fit(linear_fit, classic_studiot, classic_studio)
est_std_devs6 = np.sqrt(np.diag(est_covs6))
[k7,b7], est_covs7 = curve_fit(linear_fit, compact_studiot, compact_studio)
est_std_devs7 = np.sqrt(np.diag(est_covs7))
[k8,b8], est_covs8 = curve_fit(linear_fit, diamond_studiot, diamond_studio)
est_std_devs8 = np.sqrt(np.diag(est_covs8))
[k9,b9], est_covs9 = curve_fit(linear_fit, premium_studiot, premium_studio)

#calculate the prices of each type of room with fit coeffient
fit_number1 = linear_fit(np.array(non_ensuite_2t), k1, b1)
fit_number2 = linear_fit(np.array(ensuite_2t), k2, b2)
fit_number3 = linear_fit(np.array(ensuitelarge_2t), k3, b3)
fit_number4 = linear_fit(np.array(ensuite_4t), k4, b4)
fit_number5 = linear_fit(np.array(ensuite_5t), k5, b5)
fit_number6 = linear_fit(np.array(classic_studiot), k6, b6)
fit_number7 = linear_fit(np.array(compact_studiot), k7, b7)
fit_number8 = linear_fit(np.array(diamond_studiot), k8, b8)
fit_number9 = linear_fit(np.array(premium_studiot), k9, b9)

#plot the line of best fit
ax.plot(non_ensuite_2t, fit_number1, color="brown", label=r"2 bedroom non_ens
uite")
```

```
ax.plot(ensuite_2t, fit_number2, color="darkorange", label=r"2 bedroom ensuit
e")
ax.plot(ensuitelarge_2t, fit_number3, color="greenyellow", label=r"2 large be
droom ensuite")
ax.plot(ensuite_4t, fit_number4, color="olive", label=r"4 bedroom ensuite")
ax.plot(ensuite_5t, fit_number5, color="teal", label=r"5 bedroom ensuite")
ax.plot(classic_studiot, fit_number6, color="violet", label=r"classic studio"
)
ax.plot(compact_studiot, fit_number7, color="skyblue", label=r"compact studio
")
ax.plot(diamond_studiot, fit_number8, color="lightcoral", label=r"diamond stu
dio")
ax.plot(premium_studiot, fit_number9, color="turquoise", label=r"premium stud
io")
ax.legend()
plt. savefig('plot.png', dpi=300)
files.download("plot.png")

#collect slopes, uncertainties of fitted slope into lists
slopes = [k1,k2,k3,k4,k5,k6,k7,k8,k9]
est_std_devs = [est_std_devs1,est_std_devs2,est_std_devs3,est_std_devs4,es
t_std_devs5,est_std_devs6,est_std_devs7,est_std_devs8]

#calculate the percentage change of rent based on rent of 2021
data_2021 = df.loc[df['year'] == 2021 ,'Rent_week'].to_numpy()
percentage_increase = slopes/data_2021
print(percentage_increase)
```