Housing St Andrews | Vertically Integrated Project

Candlemas Semester 2024

Progress Report 2

Final Progress Report

Submitted by:	Georgi Butch, Elena Chesser, Tasha Delvecchio, Laurie Dewar, Marie-Claire Joyeaux, Bahrathi Keeping, Aniket Khurana, Phoevos Kreps, Tara Nair, Kieran Pirie, Finn Watson, Marcin Weremczuk, Gabriella Yuschenkoff, Ajda Vili Kapelj Zupan
Submitted to:	Dr. Luc Bridet and Dr. Sumedh Dalwai
Submission Date:	05.04.2024
Word Count:	26,702

Table of Contents

List of Abbreviations

CASH	Campaign for Affordable Student Housing
CSARA	Confederation of St Andrews Residents’ Associations
FOI	Freedom of Information
HMA	Housing Market Area
HMO	Houses of Multiple Occupation
LHS	Local Housing Strategy
ONS	Office for National Statistics
UK	United Kingdom
VIF	Variance Inflation Factor

Abstract

Student towns feature high-turnover housing markets with unique characteristics, including a transient student population, seasonal migration, and the demand-side driver of university enrolment. In the student town of St Andrews, Scotland, recent increase in rent prices and studentification have prompted housing affordability concerns from students and resident groups alike, leading Fife Council to introduce a policy in 2019 aimed at restricting the growth of Houses in Multiple Occupation (HMO). HMOs are properties that are rented to three or more unrelated individuals, are required by law to be licensed by the local authority and, in the context of St Andrews, are primarily occupied by students. With the goal to explore how this 2019 HMO Overprovision Policy affects the housing market, the statutory register of licensed HMO properties in St Andrews is analysed, revealing a significant volatility of active licenses between 2020 and 2024, ranging from 547 to 1,059. Further, wanting to understand key supply-side factors influencing rent prices, a hedonic pricing regression on property characteristics is conducted. Using web-scraping to build a dataset covering the St Andrews rental market from 2012 to 2024, monthly rent per room is found to increase by £3.69 each year, after adjusting for inflation. Additionally, a demand-supply model of the town's private rental market is calibrated. By assuming a 1.5% yearly increase of supply in the hypothetical case of no HMO Overprovision policy implementation, this counterfactual analysis deduces that 29% of the increase in average rent between the years 2019 and 2023 stems from the policy's introduction as a supply shock, while 71% is attributed to demand factors such as the growing student population. Lastly, to investigate further demand-side factors that affect students' rental decisions, two regressions are conducted on 716 responses collected by a housing survey conducted in March 2024. When examining the effect of fee status, funding source, and ethnicity on rent paid, it is observed that international fee-paying students pay £100.32 higher rent per month than UK fee-paying students, and family-funded students pay £27.64, £89.02, and £171.95 more than those who have student loans, fund themselves, or have accommodation scholarships, respectively. Ethnicity is found to have a statistically insignificant effect on rent paid. This progress report provides a transparent documentation of results produced on the analysis of the student town of St Andrews. This progress report describes the research progress of the Housing in St Andrews Vertically Integrated Project (VIP) during the Candlemas semester of 2023/24.

Keywords: Studentification, Houses in Multiple Occupation (HMO), HMO Overprovision Policy, Students, Student Housing

1. Introduction

In recent years, urban areas across the globe have witnessed a phenomenon known as studentification, characterised by the influence of students in residential neighbourhoods adjacent to educational institutions, driving demographic and spatial change, particularly so in British towns (Smith, 2005). Studentification as a social issue has attracted a growing body of research over the past years (see Allinson, 2006; Gregory & Rogerson, 2019; Hubbard, 2008; Munro et al., 2009; Sage et al., 2012; Smith, 2005; Smith & Holt, 2007) and has become of interest in recent literature for having the potential to significantly increase or inflate the property market (Smith, 2005). Especially, the significant change to the resident profile of university cities in the United Kingdom (UK) is being associated with an oversupply of student houses in multiple occupation (HMO; Munro et al., 2009), leading to privately rented HMOs overtaking owner-occupied housing stock and hence contributing to a rise of areas which are solely student-centred (Allinson, 2006). Thus, scholarly interest in addressing the topic of the impact of the implementation of HMO policies on UK university towns has grown in recent years. Nonetheless, the university town of St Andrews, situated on the east coast of Scotland in Fife, has never been in focus in previous literature but is of particular interest, as it is exemplary for a rise of a student-centred town where the total population is estimated at 18,762 as of 2024 (World Population Review, 2024), from which in 2023 a total of 10,468 were students, numbers which represent a growth of 16.5% in student population since 2018 (VIP Project, 2023). Even though the increase in St Andrews lies within the UK's national average of student population growth of 16.5% (an increase of 406,075 students) between 2018/19 and 2021/22 (HESA, 2023), the growth of student numbers in the UK prompts an issue for housing markets, residents, and students alike in all university cities across the UK.

Consequently, Fife Council's reaction to address concerns over student housing outpricing locals has come in form of setting an overprovision policy into place in 2019 that restricts the increase of HMO-licensed properties in St Andrews (Fife Council, 2023), thereby presenting a new supply constraint to the market. As such, it is crucial to explore the implications of the HMO overprovision policy on the housing market in St Andrews and, specifically, how this new policy affects students and student housing availability. The analysis has four key findings which are brought forward by the four different teams of the project: (i) Register, (ii) Supply, (iii) Demand, and (iv) Survey.

First, with the goal to explore how the new HMO Overprovision Policy affects the housing market, especially changes in HMO licenses and the response of property owners in response to the new policy, the statutory register of licensed HMO properties in St Andrews is analysed. A significant volatility in the number of active licenses between 2020 and 2024, ranging from 547 to 1,059, is identified which can be reconciled with Fife Council's cited stable supply when pending renewal applications are considered as active supply. Additionally, HMO licenses are tested for differential attrition based on property characteristic including occupancy, property value, and distance to town. No statistically significant effects are found at the 5% significance level, indicating that landlords do not selectively sacrifice HMO licenses based on the studied characteristics. This finding challenges findings of previous studies which have documented property characteristics such as property size, location, amenities, and quality to be shaping market responses (Goodman & Thibodeau, 1998; Saiful Islam & Asami, 2010; Zietz et al., 2006). Chapter 4, Register Team evaluates in more detail.

Second, given the rising rental prices in St Andrews over the last decade, a hedonic pricing regression is applied to illuminate the key supply-side factors that determine rental prices. Using web-scraping to build a dataset covering the St Andrews rental market from 2012-2024, three key findings emerge: (i) Monthly rent per room has increased by £3.69 each year, equivalent to monthly rent increasing by £44.28 over the 12 years after adjusting for inflation, underlining findings in literature, namely that mean rental prices are seen to be elevated by student HMOs (Sage et al., 2012). There is (ii) a U-shaped relationship between number of rooms in a property and rent price per room; rent per room is highest for 1-bed properties and declines to a minimum for 3-bed properties, and then increases again up to 6-bed properties. Lastly (iii), rent per room declines by £24.05 for every kilometre increase in walking distance from the centre of town (defined as Tesco Express at 138-140 Market Street), suggesting that moving from the 75th percentile (3km) to the 25th percentile (1.3km) distance to town increases monthly rent per room by £31.27. Chapter 5, Supply Team evaluates in more detail.

Third, being interested in understanding the factors behind the rise in average rental prices following the implementation of the HMO Overprovision Policy, another focus is set on discerning the extent to which the policy change contributes to the rent increase in St Andrews, as without a corresponding student hall infrastructure extension, student number growth increases demand in the private rental market and consequently pressures rent levels upwards. Using a linear supply and demand model, specific to this town's private rental housing market of HMO-licensed properties, it is calibrated in accordance with the market data gathered through web scraping of property listing websites. The average rent increase between the years 2019 and 20203 is then analysed by breaking it down into the impact of the HMO Overprovision policy (supply shock) and the influence of the expanding student population (demand shock), assuming a 1.5% yearly increase of supply in the hypothetical case of no HMO Overprovision policy implementation. Further, to isolate the demand shock, a constant student population instead of the observed growth between 2019 and 2023 is presumed. Through this counterfactual analysis, it is determined that 29% of the rent increase is attributed to the HMO Overprovision policy, while 71% is linked to the rise in student numbers. In total, the modelled rent per room increase from 2019 to 2023 is found to be 9.84%. Chapter 6, Demand Team, evaluates in more detail.

Fourth, to investigate further demand-side factors that affect students' rental decisions, two regressions are conducted on 716 responses collected by the March 2024 housing survey. For the first regression, it is hypothesised that of the results collected in the survey, the three main contributing factors are fee status, funding source, and ethnicity. When examining the effect of each of these factors on the rent paid, it is observed that international fee-paying students pay £100.32 higher rent per month than UK fee-paying students, and family-funded students pay £27.64, £89.02, and £171.95 more than those who have student loans, fund themselves, or have accommodation scholarships, respectively. Ethnicity is found to have a statistically insignificant effect on rent paid. Chapter 3, Survey Team, evaluates in more detail.

Last but not least, data gathering and methodology from the previous teams are improved to ultimately support and extend previous findings (VIP Project, 2023). Therefore, this progress report is divided as follows: Firstly, an introduction about the HMO policy and background to the situation in St Andrews is given, followed by the core part which sets into focus the different sub-teams of the project, namely Survey, Register, Demand, and Supply. In their respective chapters, the sub-teams delve into their work by presenting their data, chosen methodology, and results. Finally, after every team discusses their outcomes, general lessons inferred from working on the project that apply to everyone and the process of learning are put forward. By doing so, this project contributes to the current strand of academic literature that focuses on studentification and HMO policies, and extends existing findings by focusing on the student town of St Andrews.

2. Background on the HMO Policy

The management of Houses in Multiple Occupation (HMO) stands as a critical juncture where legislative frameworks intersect with socioeconomic realities and community dynamics. St Andrews, nestled along the coast of Fife, Scotland, serves as a microcosm of the broader challenges facing municipalities grappling with housing affordability, community integration, and studentification.

Insights from HMO Licensing Frameworks in Scotland

HMO licensing policies in Scotland are rooted in the legislative mandate of the Civic Government (Scotland) Act 1982 (Civic Government, 1982) and the Housing (Scotland) Act 2006 (Scottish Parliament, 2006), aiming to ensure the safety and welfare of residents in shared accommodation arrangements. In particular, this legislation puts forward an HMO licence to be mandatory for any rental accommodation and further specifies that within the meaning of the 2006 Act, a living accommodation is defined as an HMO if it (i) is occupied by three or more people from three or more families, (ii) is occupied by them as if it were their primary residence, and (iii) provides shared basic amenities (Scottish Parliament, 2006). Moreover, this enactment also covers other types of residential accommodation, with specifically student halls of residence being included. Once an HMO license is granted, it is in effect for three years, however, under Section 131A of the Housing (Scotland) Act 2006 and certain evaluations of (i) location, (ii) condition, (iii) amenities, and (iv) safety, the Council may deny allowance of an HMO permit in locations where it considers that there is an oversupply of HMOs (Scottish Parliament, 2006).

HMO Overprovision Policy Dynamics in St Andrews

The concept of “overprovision” lies at the core of discussions surrounding HMO licensing policies. Defined loosely as an excess of HMO properties within a locale, the notion of overprovision embodies divergent interpretations amongst policymakers, residents, and industry stakeholders and has been implemented in different forms by several local authorities in Scotland already, including Aberdeen, Dundee, and Stirling (Aberdeen City Council, 2016). Within the St Andrews context, the Fife Council HMO overprovision policy reflects attempts to address concerns over housing affordability and student housing outpricing locals and further driving studentification (Fife Council, 2023). Therefore, a no growth strategy in the number of HMO licenses was set into place on April 11th, 2019. Under the policy, landlords wishing to renew a license about to expire must apply for a renewal before the expiration of their old license and the authorities must, by law, make a decision within 12 months of the application date, yet, pre-existing licenses already in force are not affected (Fife Council, 2022).

With the goal to evaluate the policy and its no growth strategy in St Andrews, a 2023 policy review was conducted by Fife Council which found that the policy has reduced the number of HMOs from 1,046 HMOs in March 2019 to 1,029 HMOs in February 2023, proving the no growth strategy to have been successful (Fife Council, 2023). The no growth goal in the number of HMO licenses stems from the fact that a considerable majority of 86% of the total of 1,219 HMO licenses across Fife were located in St Andrews in the year 2019, with St Andrews’ housing stock being accounted for 15% by HMO-licenced properties (Fife Council, 2023). This disproportionate concentration of HMOs in St Andrews not only reflects the unique housing landscape of the town, but also underscores the challenges associated with studentification and housing affordability, prompting regulatory interventions aimed at preserving the residential diversity of the community (St Andrews QV, 2018).

Responses to HMO Policy Implications in St Andrews

Whereas the no growth strategy has achieved its objective, the HMO overprovision policy has also prompted a decrease in available accommodation for St Andrews university students (a decrease of seventeen HMO licenses and thus accommodation for 124 occupants), leading to the University of St Andrews pointing out the issue of student homelessness (Fife Council, 2023). Consequently, not only the university has announced to build more student halls and provide additional accommodation options outside of St Andrews, but also the Council has opted to introduce more flexibility to their targeted approach and renew the policy, promising to approve for up to fifteen new HMO licenses to be issued to private properties that are managed through the university to help counteract student homelessness (Fife Council, 2023). Nevertheless, the restriction of rental housing supply in St Andrews has increased tensions between different stakeholders involved, giving rise to activist groups and campaigns such as the Campaign for Affordable Student Housing (CASH) or the Confederation of St Andrews Residents' Associations (CSARA; St Andrews QV, 2022), ultimately making it important to explore the issue of HMO licensing in St Andrews and the development of studentification in St Andrews in more detail.

3. Survey Team

In 2022/2023, the Survey team created a Qualtrics survey to gather housing data from students consisting of 14 questions with multiple parts. Over the course of four weeks, this survey received 635 responses, of which 540 were fully completed. In their limited time frame, the previous team carried out some descriptive statistics and evaluated the demographics of survey respondents. Some initial regressions investigating whether people of different ethnicities and genders pay different rent were explored as well. In semester one of 2023/24, the Survey team was redirected to become Outreach and no progress was made on survey analysis.

This semester, the Survey team furthered the analysis of the previous survey's data and ran a second survey to collect additional useful housing data. The team comprises of four students: Elena Chesser, Laurie Dewar, Aniket Khurana, and Gabriella Yuschenkoff. Our aims were to clean and analyse the previous survey data, run a second survey, and complete preliminary analysis on the new data. In the first half of the semester, we divided into two teams: new survey and old survey. The new survey team, Aniket and Elena, endeavoured to rewrite the previous questions and design a new survey. The old survey team, Gabriella and Laurie, spent time cleaning and analysing the previous survey data to identify areas for improvement. Once the survey was released, we reconfigured the team structure with Gabriella and Laurie working on promotion efforts and Elena and Aniket focusing on Qualtrics. We received 716 responses to the new survey from undergraduate students and postgraduate students combined. Of these 716 responses, 607 were fully complete.

3.1 22/23 Survey

a. Data

The anonymised responses from the previous survey are stored in the "All Respondents Cleaned" sheet in the excel file titled "HMO Caps in St Andrews VIP Survey Results". It contains 540 observations. We used this data in our analysis, building on the previous team's STATA code with new cleaning procedures and regressions (See Appendix 1). To complete meaningful analysis, we needed to standardise the data. This entailed: extracting responses that were variations of "n/a"; creating loops for each variable to check frequency of responses and standardize n/a responses; converting rent into a numerical variable rather than string; fixing a year of study typo; simplifying the variable names; Assigns dummy variables to qualitative responses (fee status, bedroom number, ethnicity and who is responsible for paying the rent) (For Stata code, see Appendices 2 and 3).We also found that some respondents reported their yearly data instead of monthly, which skewed the rent averages upwards. To correct this, after being converted from a string to a number the rent data is checked and values above £3000 were assumed to be 9-month halls contracts and thus divided by 9. The new variable rent is used for regressions. (Appendix 4)

b. Methods

Using data gathered from the 2023 survey an Ordinary Least Squares (OLS) model was constructed to explain the relationship between financial characteristics and rent. For simplicity we assume a linear relationship between the independent variables and rent. We also assume that for the characteristics under study the Gauss Markov assumptions hold. This is optimistic, particularly the orthogonality assumption as other factors, such as expected future earnings (and thus whether you chose to take out the maximum loan), would also affect one's property choices. We hypothesised that the financial circumstances of the student: their fee status and who was responsible for rent payments would have had a significant impact on their rental budget. We controlled for the bedroom numbers of the properties under the assumption that there would be a negative linear correlation between price and bedroom number (i.e. living in a studio would be the most expensive to rent). We also controlled for ethnicity to see if, in a majority white town, people of colour experienced a distinct housing market.

We wanted to test the assumption that properties on the three streets and immediate surroundings (Market Street, South Street and North Street) have higher rents than properties further afield. Tesco Express on Market Street was designated as the centre of town – whilst this may not be the ideal location for all students, the area has the highest concentration of businesses and university buildings. The location data of respondents was stored as postcodes, and so we needed to create a numerical variable. To do so, firstly duplicates were removed, and the postcode variables were saved into a "postcode_survey" dataset which is updated throughout the process. The main dataset is then used. The postcode was extracted from the address variable (or postcode variations e.g. DD for Dundee). In cases where postcode was not given, the dataset of all St Andrews postcodes is paired with the address. This pairing is done through 'fuzzy matching': identifying the postcode from the street name whilst allowing for slight deviances. A similarity of a score greater than .7 is set as the cut off and invalid postcodes are set to missing. After the matching process the relevant dataset was reordered and unnecessary observations that weren't matched but were contained in the second dataset were removed. Having merged the postcodes, we can now add the *DistancetoTown* data (the distance from Market Street Tesco retrieved from Google Maps) to our observations. This process is described in Appendix 11.

c. Results

When testing our assumption that the properties closest to the centre are the most expensive, we found that the correlation between distance to town and rent was only -0.1313.

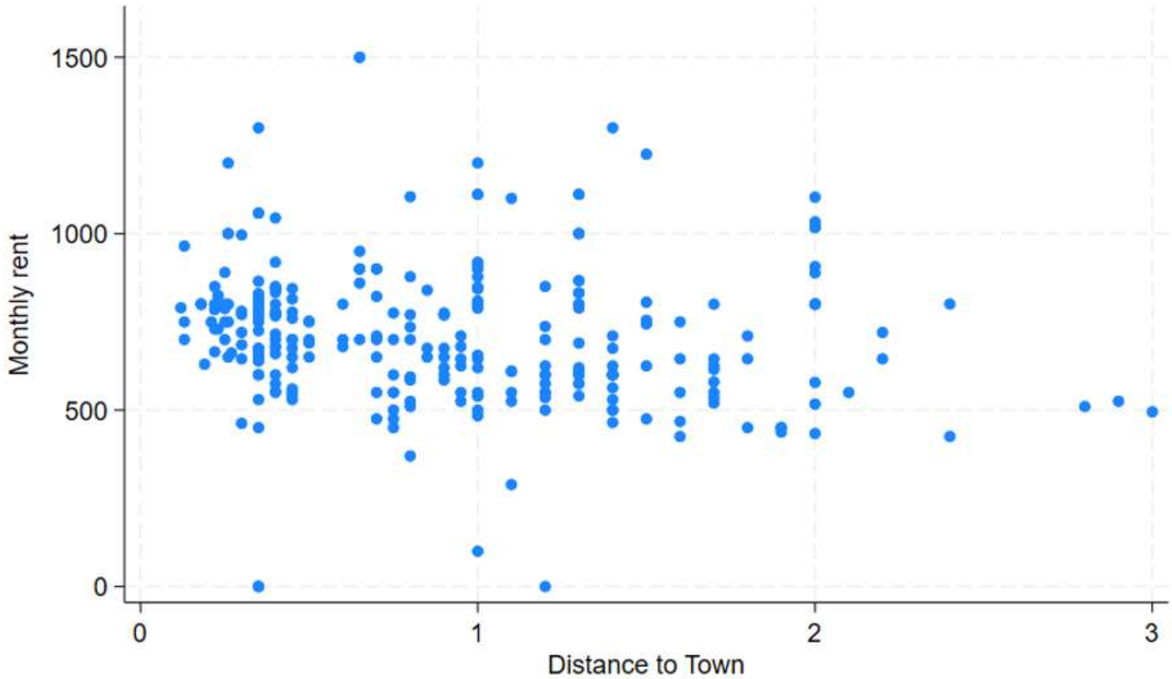


Figure 1
Scatter Plot of Individual Monthly Rent and Distance to Town.

To further test the impact of *DistancetoTown* on rent we set up a regression consisting of only these two variables (Figure 2).

```
. regress rent DistancetoTown

      Source |       SS           df       MS      Number of obs   =       295
-------------+----------------------------------   F(1, 293)       =      5.14
       Model |   194106.68         1   194106.68   Prob > F        =    0.0241
    Residual |  11057237.3       293  37738.0114   R-squared       =    0.0173
-------------+----------------------------------   Adj R-squared   =    0.0139
       Total |    11251344       294  38269.8776   Root MSE        =    194.26

------------------------------------------------------------------------------
          rent | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
---------------+----------------------------------------------------------------
DistancetoTown |  -43.77144   19.30012    -2.27   0.024    -81.75588   -5.786992
         _cons |    751.787    20.8234    36.10   0.000     710.8047    792.7694
------------------------------------------------------------------------------
```

Figure 2
Regression of *DistancetoTown* on *Rent*

Whilst *DistancetoTown* was indeed significant in explaining rent (p-value of 0.024), the R-squared value was only 0.0173. This suggests that it is only a small proportion of the overall rent. However, the coefficient of ~ -43.77 on *DistancetoTown* provides quantitative information as to how much rent decreases as the distance to town from a respondent's property increases.

We then set up a regression for monthly rent using fee status, ethnicity, number of bedrooms and who pays your fees as the variables. The constant in the regression represents what the expected individual monthly rent is for a white, Scottish, family-funded student living in a studio.

```
. regress rent RUKfees otherfees intlfees twobedrooms threebedrooms fourbedrooms fivebedrooms sixormorebedrooms nonwh
> ite otherfunding ownincome nafunding studentloans accomgrant

      Source |       SS           df       MS      Number of obs   =       508
-------------+----------------------------------   F(14, 493)      =      8.89
       Model |   4488649.4        14  320617.814   Prob > F        =    0.0000
    Residual |  17772630.7       493  36049.9609   R-squared       =    0.2016
-------------+----------------------------------   Adj R-squared   =    0.1790
       Total |  22261280.1       507  43907.8503   Root MSE        =    189.87

------------------------------------------------------------------------------
             rent | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
------------------+----------------------------------------------------------------
          RUKfees |   9.420489    22.6479     0.42   0.678    -35.07781    53.91879
        otherfees |  -11.99586   66.45617    -0.18   0.857    -142.5681    118.5764
         intlfees |   58.74081   22.36907     2.63   0.009     14.79034    102.6913
      twobedrooms |   -165.398   23.23272    -7.12   0.000    -211.0454   -119.7507
    threebedrooms |  -163.8191   30.22271    -5.42   0.000    -223.2003   -104.4379
     fourbedrooms |  -181.8276   27.89513    -6.52   0.000    -236.6356   -127.0196
     fivebedrooms |  -113.2559   30.29059    -3.74   0.000    -172.7705    -53.7413
sixormorebedrooms |  -127.3639   43.85719    -2.90   0.004    -213.5339   -41.19379
         nonwhite |  -7.989152   21.21972    -0.38   0.707    -49.68138    33.70308
     otherfunding |  -163.4635   43.09615    -3.79   0.000    -248.1383   -78.78871
        ownincome |  -30.20516   30.91146    -0.98   0.329    -90.93961    30.52929
        nafunding |   12.26931   52.39881     0.23   0.815    -90.68321    115.2218
     studentloans |   -40.3175   22.72548    -1.77   0.077    -84.96825     4.33325
       accomgrant |   63.84204   73.43528     0.87   0.385    -80.44268    208.1268
            _cons |   814.0196   20.99621    38.77   0.000     772.7665    855.2727
------------------------------------------------------------------------------
```

Figure 3
Stata Generated Rent Regression.

Many variables in the regression were found to be insignificant at the 5% and 10% hypothesis level, however all bedroom variables were found to be significant. However, the large negative coefficients tied to flats with less bedrooms are unexpected. One would assume that with more bedrooms the rent per person would generally decrease. The regression demonstrates that six or more bedrooms decrease the rent by 127.36 while two bedrooms only decreases the rent by 165.40, implying that students living in properties with more bedrooms and more flatmates will pay more than a student living in a property with less bedrooms and less flatmates.

There is no evidence of multicollinearity in the model or incorrect functional form. We also fail to reject the null of homoskedasticity with a p-value of 0.01 for the White test.

```
. *test heteroskedasticity
. estat imtest, white

White's test
H0: Homoskedasticity
Ha: Unrestricted heteroskedasticity

   chi2(71) =  99.51
Prob > chi2 = 0.0144

Cameron & Trivedi's decomposition of IM-test
```

Source	chi2	df	p
Heteroskedasticity	99.51	71	0.0144
Skewness	22.97	14	0.0608
Kurtosis	9.98	1	0.0016
Total	132.46	86	0.0010

Figure 4
Test for Heteroskedasticity

We hypothesised that international students would pay the highest rent out of the fee status categories due to the criteria of needing to afford fees between £20,000-£36,000. We also hypothesised that Rest of UK (RUK) students would pay the second highest rent, given that they must budget accommodation on top of £9,250 a year for tuition fees. This was confirmed by the survey results. The coefficient for RUK fee status was 9.420, which tells us that RUK students are expected to pay £9.42 more per month than Scottish students. For international students, the coefficient was 58.74, indicating a difference of £58.74 between the rents of Scottish and International fee-payers.

Yet only the dummies of non-white, own-income, international are independently significant at the 0.05 level and there is not overwhelming evidence to rule out the possibility of incorrect functional form and omitted variable bias. Non-white students are expected to pay £7.99 less monthly rent than white students. The 'non-white' ethnicity group was created to test the assumption that housing circumstances could be negatively affected by belonging to a member of a non-white ethnic minority. Grouping non-white ethnicities also help us analyse a much larger sample as most individual ethnicities have small sample sizes (for example, only 8 respondents identified themselves as “Black/African/Caribbean”, and 26 identified as “Mixed two or more ethnic groups”) - limiting our ability to draw reliable conclusions for individual non-white groups.

Students using their student loans to pay for their accommodation are modelled to pay £40.32 less than those whose families pay their rent. Those with an accommodation grant are expected to pay £63.84 more than family funded students. Finally, those paying with their own income are expected to pay £30.21 less than family funded students. The fact that some factors are statistically insignificant may be caused by the fact that many students use more than one source of income to pay their rent; for example, a student may have a small maintenance loan in addition to parental contributions. Without asking intrusive personal finance questions, it would be difficult to capture the precise impact of different factors.

The number of bedrooms, intuitively, has a large impact on an individual monthly rent. On a descriptive level, studios were found to be the most expensive property type. Each relative to studios, five-bed properties were the second most expensive at £113.26 less per person, six or more-bed properties came in at £127.36 less per person, 3-bed properties at £163.82 less per person, 2-bed properties at £165.40 less per person and four-beds were the cheapest at £181.83 less per person. We included these in the regression to separate an important predictor of rent from the fee category, ethnicity and funding source factors.

We hypothesized that first-year responses were confounding the regression because we believed that allocation of first years into halls was arbitrary. To test this, we excluded first year students from the regression to see if they were blurring the model with random halls allocation. However, we found that removing first-years from the model does not significantly change the coefficients and the adjusted R-squared only changed from .179 to .171. This suggests that we should reject our hypothesis and conclude that our factors are also important for first-year hall selection. Anecdotally, this could be because students paying their rent with their own income may not choose expensive catering plans, which added just under £3000 to the 9-month contracts.

```
. regress rent RUKfees otherfees intlfees twobedrooms threebedrooms fourbedrooms fivebedrooms sixormorebedrooms nonwhite other
> funding ownincome nafunding studentloans accomgrant if year_num!=1
```

Source	SS	df	MS			
				Number of obs	=	378
				F(14, 363)	=	6.57
Model	3415588.21	14	243970.586	Prob > F	=	0.0000
Residual	13473066.2	363	37115.8849	R-squared	=	0.2022
				Adj R-squared	=	0.1715
Total	16888654.4	377	44797.4918	Root MSE	=	192.65

rent	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
RUKfees	17.93855	27.99556	0.64	0.522	-37.1153	72.99239
otherfees	-4.871548	68.31644	-0.07	0.943	-139.2172	129.4741
intlfees	64.79959	26.46639	2.45	0.015	12.75289	116.8463
twobedrooms	-167.7798	27.98517	-6.00	0.000	-222.8132	-112.7464
threebedrooms	-164.1829	34.2761	-4.79	0.000	-231.5875	-96.77822
fourbedrooms	-179.6223	32.53576	-5.52	0.000	-243.6045	-115.64
fivebedrooms	-113.7122	34.51302	-3.29	0.001	-181.5828	-45.84165
sixormorebedrooms	-130.2693	47.54084	-2.74	0.006	-223.7593	-36.77924
nonwhite	12.54803	24.797	0.51	0.613	-36.21577	61.31184
otherfunding	-213.2329	48.76215	-4.37	0.000	-309.1247	-117.3411
ownincome	-40.72661	35.05892	-1.16	0.246	-109.6707	28.21749
nafunding	-36.39967	63.33683	-0.57	0.566	-160.9528	88.15351
studentloans	-48.04159	26.96457	-1.78	0.076	-101.068	4.984794
accomgrant	38.73923	98.90284	0.39	0.696	-155.7552	233.2337
_cons	812.5145	29.33658	27.70	0.000	754.8235	870.2055

Figure 5
Regression as above but excluding First Year Students

3.2 2024 Survey

a. Data

The anonymized responses for the new survey are stored in "All Responses" of the excel file "Anonymized Data 27.03" and has 716 observations. We used STATA to analyse this data, building on the code we created to evaluate the 22/23 survey's responses. Because we standardized the response options in the survey (i.e. not allowing free "type your answer here" responses), we did not need to clean as much of the data. A value of “63708” was changed to “637.08” in the rent responses as it was discovered that the respondent used a comma instead of a period to denote pence. Responses to “How much has your rent increased?” included “+” or worded answers such as “by ‘x’ amount in ‘y’ years”, however we did not clean this variable. We opted to only clean the variables we were planning to use in the regression, to save time and prioritize confirming our previous results.

b. Methods

Data was collected using a newly drafted survey created in Qualtrics. Once the survey was closed to respondents, the data was transformed into an excel file titles “Anonymized Data 27.03” and converted into a do file for STATA analysis. We knew based off the previous survey analysis we conducted that fee status and income responsibility were two variables we would focus on within this analysis to understand how a student’s financial situation impacts their rent.

Because undergraduates and postgraduates have different fee status classifications we grouped “Rest of UK,” “Scottish,” and “Islands” together to create the variable *UK* denoting the fee status of residents of the UK. This was done by generating a dummy variable labelled *UK* and replacing *UK* = 1 if variable *feestatus* was equal to "RUK (England, Wales, Northern Ireland, and Republic of Ireland)","Islands (Channel Islands and Isle of Man)", "Home (Scotland)", or "Home (UK, Channel Islands, and Isle of Man)"). This allows us to use both postgraduate and undergraduate responses to fee status in our regression to accurately explain fee status’s impact on *rent* for all student types.

A second grouping labelled *donotrent* was created using a grouping of responses indicating students who lived in properties managed by someone living in the household or a relative of someone living in the household. Respondents that indicated living in properties managed by themselves or their family were also grouped into *donotrent*. To do this, we employed the same system used for creating grouped variable *UK*. We created a dummy variable labelled *donotrent* and replaced *donotrent* = 1 if *Whoownsmanagesyourproperty* was equal to "My family or me" or "Someone living in the household or a relative of someone in the household." Using a grouping will allow us to analyse the impact of property management type on an individual’s monthly rent and identify whether living in a personal property, with a relative, or someone who owns the property decreases *rent.*

Preliminary testing revealed an international fee status (variable *international*) has a positive correlation with rent while a grouped UK fee status (variable *UK*) has an equivalent but negative correlation with rent. This was to be expected considering Scottish students are not required to pay tuition while international students are required to pay above £20,000 in tuition fees and have likely prepared in advance so can afford a larger budget.

```
. corr rent international UK
(obs=410)

             |     rent intern~l       UK
-------------+---------------------------
        rent |   1.0000
internatio~l |   0.2062   1.0000
          UK |  -0.2062  -1.0000   1.0000
```

Figure 6
Correlation of Fee Statuses and *rent*

We found that there is positive correlation between *rent* and *familyfunded* but negative correlations betwee*n rent* and all other types of funding (i.e. external funding, scholarships, student loans, self-funding), implying that students whose family members pay their rent live in pricier accommodation.

```
. corr rent scholarship selffunded studentloans extfunding familyfunded
(obs=410)

             |     rent schola~p selffu~d studen~s extfun~g family~d
-------------+------------------------------------------------------
        rent |   1.0000
 scholarship |  -0.1109   1.0000
  selffunded |  -0.0873  -0.1182   1.0000
studentloans |  -0.0578  -0.1357  -0.2698   1.0000
  extfunding |  -0.0693  -0.0171  -0.0339  -0.0390   1.0000
familyfunded |   0.1959  -0.2438  -0.4847  -0.5567  -0.0700   1.0000
```

Figure 7
Correlation of "Responsible for funding" Categories and *rent*

Taking the mean of *rent* if rent > 1 (conditional to only count students who pay rent) in the new dataset reveals a slight decrease in the average rent from the 22/23 data to the 2024 data. This is surprising – recent literature suggests that the average rent in St Andrews is increasing post-Covid. However, this is not a valid comparison because the 2023 average includes catered halls contracts, which added just under £3000 to the 9-month contract in the year the data was collected, skewing the data upwards. The 2024 average excludes halls contracts and therefore is likely to be biased downwards. Whilst there may have been housing demographic changes - (the decrease observed could be due to more students choosing to live outside of St Andrews and commute to save money) future teams should evaluate these numbers critically.

```
. mean rent if rent > 1

Mean estimation                          Number of obs = 504

--------------------------------------------------------------
             |       Mean   Std. err.     [95% conf. interval]
-------------+------------------------------------------------
        rent |   740.2922   8.901269       722.804    757.7805
--------------------------------------------------------------
```

Figure 8
Mean Rent of 22/23 Survey Data

```
. mean rent if rent > 1

Mean estimation                          Number of obs = 395

--------------------------------------------------------------
             |       Mean   Std. err.     [95% conf. interval]
-------------+------------------------------------------------
        rent |   718.8052   12.60341      694.0269    743.5836
--------------------------------------------------------------
```

Figure 9
Mean Rent of 2024 Survey Data

Extensive address cleaning was not needed with the new survey responses. We used an address dataset acquired and formatted by Kieran Pirie. The first column of the dataset contained KY16 postcodes associated with St Andrews, Leuchars, Guardbridge, and other surrounding areas. The subsequent columns included the associated second half of the postcodes. This cleaned dataset was added to Qualtrics as a reference database. The postcode question asked in the survey used the dataset to enforce responses – meaning Qualtrics only counts the response as valid if the entered postcode has a match in the database. Respondents could not continue with the survey if their postcode did not match a postcode in the database. Respondents who answered “no” to the previous question “Do you live in St Andrews?” were redirected and asked to manually enter their address information. Upon successful completion of the postcode question, a follow up question with a drop-down list of addresses associated with the given postcode was shown to the participant. This lowered the friction for respondents to answer the address section (lower cost), and all postcode and address data exists in one format, removing the need for an extensive data cleaning process. Address data was then converted into a single variable *address_master*.

Once again, we used postcode merging code from our previous analysis to allow us to test whether properties on the three streets and immediate surroundings (Market Street, South Street and North Street) have higher rents than properties further afield with the newly collected data. The excel file “Postcode Continuous Variable 1” sheet “Sheet1” was imported and the same process described in the previous survey analysis was employed. The *DistancetoTown* data (the distance from Market Street Tesco retrieved from Google Maps) was added to our observations to allow us to see the impact on rent. This process is described in Appendix 11.

c. Results

i. Distance to Town: Changes from Previous Analysis

Previous analysis of the 22/23 yielded that the *DistancetoTown* variable had a correlation of -0.1313 to *rent*. Analysis of the 2024 survey data shows that the correlation between rent and *DistancetoTown* has increased to -0.1234, implying that the variable still has a negative impact on individual monthly rent, but the impact has decreased.

Values for individual monthly rent on the scatterplot have also increased. The graph demonstrates an outlier of £2,900, higher than the previous graph’s outlier of ~£1,500. The graph does indicate that higher monthly rent data points occur from properties closer to town; all recorded monthly rents above £1,000 are located within two kilometres from the centre of town (Tesco in this model). This affirms our assumption that the properties closest to the centre are the most expensive.

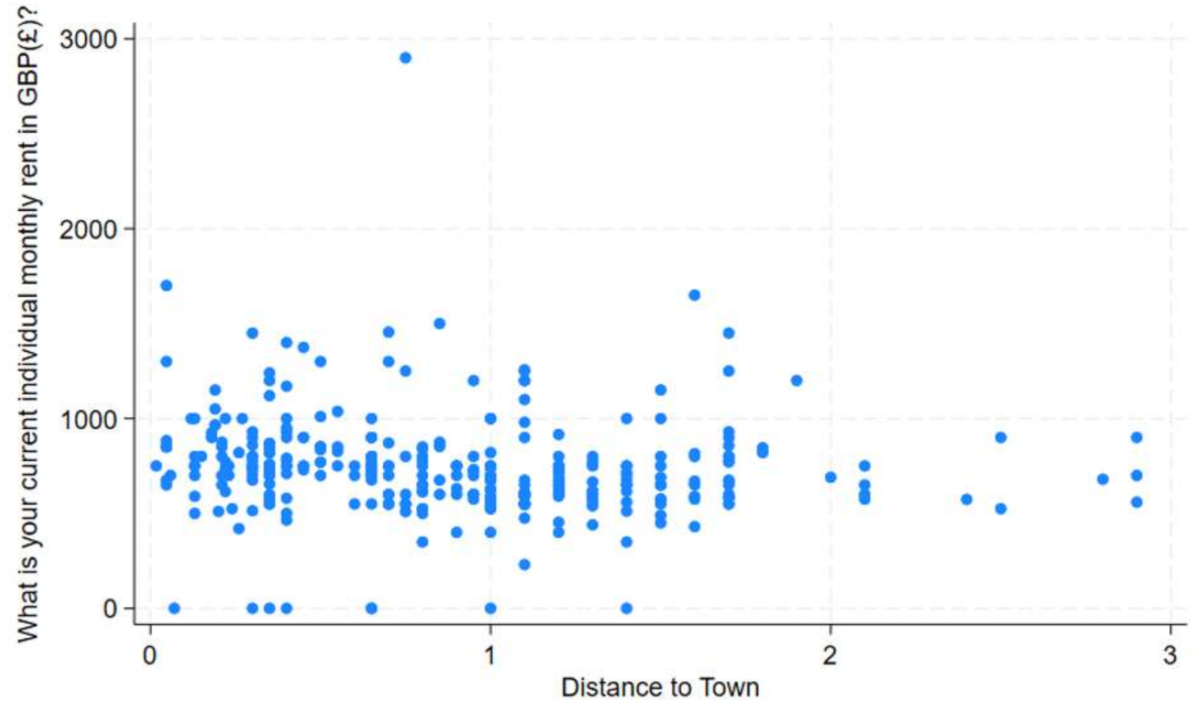


Figure 10
Scatterplot of *rent* and *DistancetoTown* with 2024 Survey Data

A regression between rent and *DistancetoTown* is used to show the relationship of distance to town on rent. The coefficient for *DistancetoTown* is ~ -59.05, indicating that living one additional kilometre from Tesco decreases rent by £59.05. This is larger than the coefficient of -43.77 on D*istancetoTown* in the previous survey data, implying the newly collected data shows an increased impact of distance on rent. With a p-value of 0.023, *DistancetoTown* is significant in describing rent at the 5% and 10% level.

```
. regress rent DistancetoTown
```

Source	SS	df	MS			
				Number of obs	=	338
				F(1, 336)	=	5.20
Model	369197.596	1	369197.596	Prob > F	=	0.0232
Residual	23867804.2	336	71035.1316	R-squared	=	0.0152
				Adj R-squared	=	0.0123
Total	24237001.8	337	71919.8866	Root MSE	=	266.52

rent	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
DistancetoT~n	-59.05316	25.90301	-2.28	0.023	-110.0057	-8.100652
_cons	792.8931	27.2701	29.08	0.000	739.2515	846.5348

Figure 11
Regression of *Rent* and *DistancetoTown* with 2024 Survey Data

While *DistancetoTown* was found to be significant in explaining rent, the R-squared was very minimal (0.0152) and we chose not include it in our regression analysis of the new survey data. We focused instead on replicating the previous regression used in analysing the 22/23 survey data to build a comparison demonstrating how the data has changed.

ii. Regression One: Explaining the average rent of students using fee status, ethnicity, and income responsibility

Building off the regression used in analysing the survey responses from 22/23 in PR1, we ran a regression of similar variables based on the new survey's responses. The variables used in this regression are *UK*, *extfunding*, *white*, *studentloans*, *selffunded*, and *scholarship*. The variable *UK* encapsulates students with a fee status falling under RUK, Scotland, and Islands. Variable *extfunding* captures respondents who use external funding to pay their expenses, *studentloan* captures students who have student loans, *selffunded* refers to students who fund their own expenses, and *scholarship* denotes respondents who use money from a scholarship to pay their fees. The variable *white* captures respondents who indicated that they are of white ethnicity. The constant in the regression represents what the expected individual monthly rent is for a non-white, international, family-funded student.

```
. regress rent UK extfunding white studentloans selffunded scholarship, robust

Linear regression                               Number of obs     =        410
                                                F(6, 403)         =      49.12
                                                Prob > F          =     0.0000
                                                R-squared         =     0.0751
                                                Root MSE          =     271.79
```

rent	Coefficient	Robust std. err.	t	P>|t|	[95% conf.	interval]
UK	-100.321	32.1938	-3.12	0.002	-163.6098	-37.03229
extfunding	-261.8022	32.31359	-8.10	0.000	-325.3264	-198.2779
white	-38.05916	38.97936	-0.98	0.329	-114.6874	38.56911
studentloans	-27.63964	36.55136	-0.76	0.450	-99.49479	44.21551
selffunded	-89.02054	38.08764	-2.34	0.020	-163.8958	-14.14527
scholarship	-171.9458	39.09759	-4.40	0.000	-248.8065	-95.08507
_cons	815.1824	36.75951	22.18	0.000	742.9181	887.4467

Figure 12

Regression of Fee Status, Funding, and Ethnicity Variables to Describe *rent*

All variables are found to be significant at the 10% hypothesis level except *studentloans* and *white*. This implies that there is not a strong explanatory relationship between rent and students who use student loans, or students who are white. Coefficients on all variables are negative, but this is unsurprising given we expected UK students who are not family-funded to live in less-expensive accommodation. Adjusted R-squared is low, implying that there is room for improvement in the regression.

White's test for heteroskedasticity reveals we reject the null of homoskedasticity with a p-value of 0.8414. However, the model is robusted to account for this.

```
. estat imtest, white

White's test
H0: Homoskedasticity
Ha: Unrestricted heteroskedasticity

   chi2(13) =   8.03
Prob > chi2 = 0.8414

Cameron & Trivedi's decomposition of IM-test
```

Source	chi2	df	p
Heteroskedasticity	8.03	13	0.8414
Skewness	2.95	6	0.8156
Kurtosis	1.42	1	0.2340
Total	12.40	20	0.9018

Figure 13
White's Test for Regression One

A RESET test F-statistic value of 2.03 is smaller than the critical value of 2.37 indicating that there is insufficient evidence to reject the null hypothesis and the model's functional form may be correctly specified. Additionally, the p-value is 0.1097, supporting the conclusion that there is not enough evidence to suggest rejecting the null hypothesis that the model is correctly specified.

```
. estat ovtest

Ramsey RESET test for omitted variables
Omitted: Powers of fitted values of rent

H0: Model has no omitted variables

F(3, 400) =   2.03
 Prob > F = 0.1097
```

Figure 14
Ramsey Reset Test for Regression One

A histogram of the residuals demonstrates a slight rightward skewed distribution. This indicates that the mean of the residual is higher than both the median and the mode. The data is mostly clustered around 0, implying there is not a large amount of variability in the data. There is, however, an outlier to the right past 2000. This is likely skewing the interpretation of the data, giving us the rightward skew versus a normal distribution.

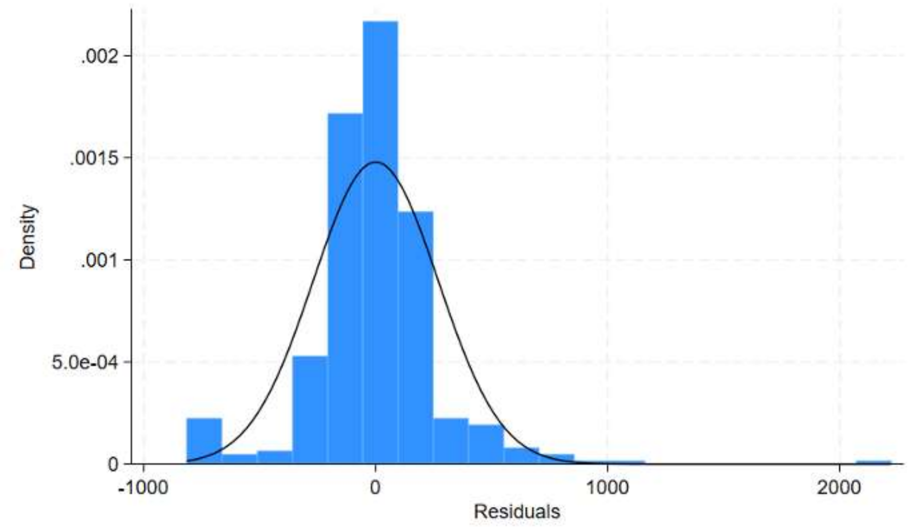


Figure 15
Histogram of Regression One Residuals with 2024 Survey Data

The results of this regression confirm our hypothesis that international students with family funding pay a higher individual monthly rent than the standard UK student with other types of funding. The regression yields that the average international, non-white, family-funded student pays £815.18 in individual monthly rent. If a student falls into the UK fee status grouping, they pay £100.32 less in individual monthly rent. This demonstrates a significant difference in rent for international students and UK grouped students, indicating that our previous findings in the analysis of the 22/23 survey were correct.

Students who use external funding pay £261.80 less and students who are self-funded (i.e. use their own income) pay on average £89.02 less. A student with a scholarship pays £171.95 less in rent. Students with student loans pays £27.64 less and students who identify as white in ethnicity pay £38.6 less, but these results were found to be insignificant. It is important to consider that the results yielded using funding variables may be biased; many students use more than one source of income to pay their rent, and this was not an option in the survey question.

iii. Regression Two: Explaining the average rent of students who do not live in private let or rented properties

The second regression explains the average rent of students who do not live in private let or rented properties. The variables used in this regression included the variables from regression one, however a new variable *donotrent* is included. This variable represents a grouping of respondents who do live in properties owned by their family/ themselves or who live with someone who owns the house or a relative of someone in the household. The constant in the regression represents what the expected individual monthly rent is for a non-white, international, family-funded student living in private let or rented accommodation.

```
. regress rent UK extfunding white studentloans selffunded scholarship donotrent, robust

Linear regression                               Number of obs     =        410
                                                F(7, 402)         =      11.81
                                                Prob > F          =     0.0000
                                                R-squared         =     0.1864
                                                Root MSE          =     255.23
```

rent	Coefficient	Robust std. err.	t	P>|t|	[95% conf.	interval]
UK	-91.37868	30.0276	-3.04	0.002	-150.4094	-32.34793
extfunding	-136.992	131.6598	-1.04	0.299	-395.8196	121.8356
white	-32.07741	36.77767	-0.87	0.384	-104.378	40.22316
studentloans	-49.96536	33.99125	-1.47	0.142	-116.7882	16.85746
selffunded	-66.76631	35.4479	-1.88	0.060	-136.4527	2.920102
scholarship	-188.595	39.90654	-4.73	0.000	-267.0466	-110.1434
donotrent	-324.0769	57.45535	-5.64	0.000	-437.0274	-211.1265
_cons	837.4866	35.06876	23.88	0.000	768.5455	906.4276

Figure 16
Regression of Fee status, Funding, Ethnicity, and Housing Type Variables

Three variables are found to be insignificant at the 5% and 10% levels: *extfunding*, *white*, and *studentloans*. In the previous regression not including variables for property type, *extfunding* was found to be significant. Most notably the coefficient on grouping variable *donotrent* is ~ –324.07. This implies that students who do not rent properties or use private landlords will pay £324.08 less on average, significantly less than students who do rent. This is expected but extremely interesting to see in numbers. R-squared on this regression is higher, indicating that the inclusion of *donotrent* increases the explanatory power of the regression. However, there is still room for improvement and the regression does not explain much of the variability of *rent*.

White's test yields a p-value of 0.9903, demonstrating that we reject the null of homoskedasticity. There is heteroskedasticity in the model, however we robust the regression above to account for this.

```
. *test heteroskedasticity
. estat imtest, white

White's test
H0: Homoskedasticity
Ha: Unrestricted heteroskedasticity

   chi2(20) =   8.22
Prob > chi2 = 0.9903

Cameron & Trivedi's decomposition of IM-test
```

Source	chi2	df	p
Heteroskedasticity	8.22	20	0.9903
Skewness	4.57	7	0.7121
Kurtosis	1.20	1	0.2732
Total	14.00	28	0.9872

Figure 17
White's Test on Regression Two

A RESET test F-statistic value of 1.25 is smaller than the critical value of 2.37 indicating that there is insufficient evidence to reject the null hypothesis and the model's functional form may be correctly specified. Additionally, the p-value is 0.2920, supporting the conclusion that there is not enough evidence to suggest rejecting the null hypothesis that the model is correctly specified.

```
. *check functional form
. estat ovtest

Ramsey RESET test for omitted variables
Omitted: Powers of fitted values of rent

H0: Model has no omitted variables

F(3, 399) =    1.25
 Prob > F = 0.2920
```

Figure 18
RESET Test for Regression Two

A histogram of the residuals for regression two demonstrates a long tail and a rightward skewed distribution. This indicates that the mean of the residual is higher than both the median and the mode. The data is mostly clustered around 0, implying there is not a large amount of variability in the data. There is, however, an outlier to the right past 2000. This is likely skewing the interpretation of the data, giving us the rightward skew versus a normal distribution. These residuals have a higher frequency around zero than in regression one and no cluster of residuals near –1000 is visible.

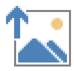


Figure 19
Histogram of Regression Two Residuals

Overall, this regression yields slightly different results from regression one. The inclusion of the grouped variable *donotrent* allows for analysis of housing type and its impact on ta respondent's monthly individual rent. On average, an international, family-funded, non-white student living in private let or rented properties pays £837.49 in individual monthly rent, slightly higher than the amount signified in regression one without the *donotrent* variable.

Students who do not live in private let or rented properties pay £324.08 less in monthly rent than their counterparts. This is likely demonstrative of commuter students living at home that do not pay rent, explaining the large decrease. Considering the decrease in mean rent observed in above analysis, this confirms thoughts that more students are choosing to commute to school from properties further from the centre of town, contributing to the decrease in mean rent observed.

3.3 Drop-off Analysis

The survey was designed to reduce the friction of completion. Structurally, this was achieved through having choice type questions instead of a text entry to allow for quicker completion of the survey. The most significant impact of this was on the address question, where instead of respondents entering their complete address, they had to enter their postcode, and a dropdown would be populated with all associated addresses. Coupled with other improvements in survey design, offer a possible explanation for the improvement in completion rate [67.3% to 84.8%]. However, this approach had a known issue, for those completing the survey on mobile, Qualtrics occasionally does not recognise the postcode even if it is entered correctly. This meant that respondents facing this issue could not proceed further as this question was required. Figure 20 highlights this in orange, where the respondent entered a valid postcode but subsequently exited the survey without Qualtrics recognising their postcode.

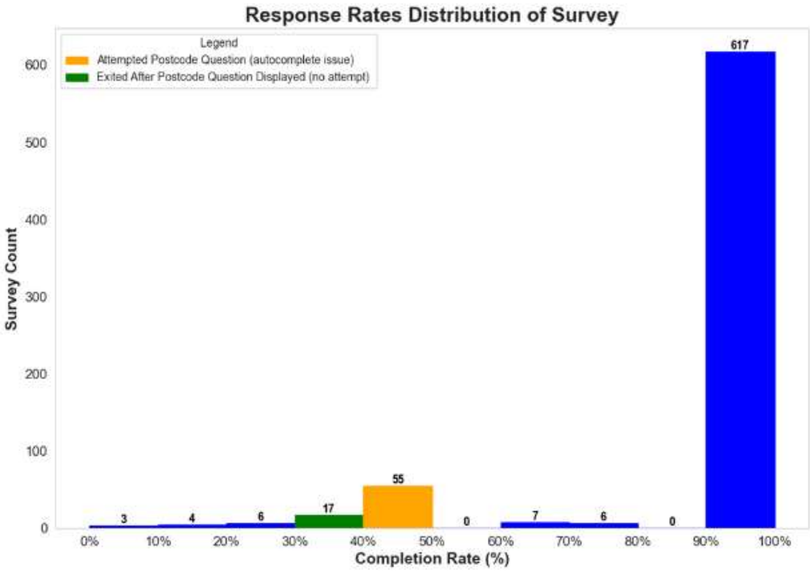


Figure 20
Graph Displaying Response Rates Distribution of Survey

Beyond that, there is no trend in drop-off points, except for a slight uptick when the postcode question was displayed. Note that this differs from earlier as respondents did not attempt the question, hence a lower completion rate as there is no recorded response. The first possible explanation is that the jump in drop-off is not extreme and could be entirely “random”. The second explanation is that as this is sensitive data which not all respondents may be comfortable sharing address, and hence left the survey.

Despite address lookup question issue, overall, we see a substantial improvement in completion rate which supports our choice to move to a choice question structure. There may be alternative approaches to the address question implementation when considering future surveys that can fix this issue without losing functionality.

3.4 Channel Analysis

As part of the survey design, a deliberate attempt was made to identify the source of every survey response. Figure 21 shows the channel breakdown of complete and incomplete survey responses. A few notes to consider: Firstly, *HSADirect* captures responses directly instigated by members of the VIP. Secondly, *Instagram* responses include any interaction on the social platform including outreach not by our Instagram account: university societies and our personal accounts. Thirdly, the responses from society emails have been clubbed into the *Societies* channel even though there is raw data for this. This was done as this data is highly fragmented, and the most significant contribution of a single society was only six responses. To identify survey sampling issues by channel, such few responses would not present a strong case.

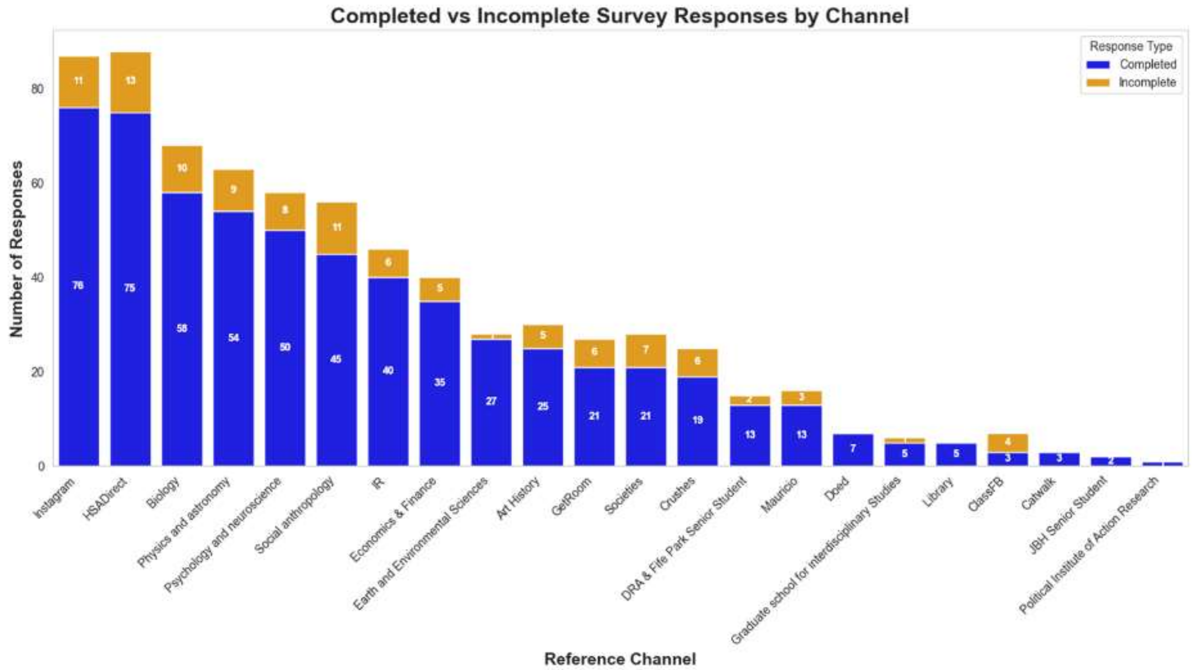


Figure 21
Graph Displaying Completed vs Incomplete Survey Responses by Channel

To avoid drawing conclusions from small samples, only the top ten channels by number of completed responses will be used to identify any sampling issues, as seen in Figure 21. The benchmarks represent the official data published by the University of St Andrews in 2019, and although the student demographics at present are likely to differ, they serve as a reference point.[1]

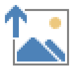


Figure 23
Graph Displaying Top 10 References by Completed Responses vs Benchmark, Sorted by % Female

Overall, responses from all but one (School of Economics & Finance) of our top-performing channels skew towards Females further than the benchmark. This is indicative of our entire survey sample also skewing in this direction. Our current data only serves as a gauge of which channels attracted the most responses, which is helpful for future survey targeting but does little to identify sampling bias on an actionable level. To do so, we need to obtain or estimate each channel's population size and breakdown. This would identify whether the skew comes from the inherit gender bias within the channels we target or whether our marketing approach leads to only a particular type of person answering. Finally, data on population size by channel would give us the conversion rate for each channel. This is useful as it can highlight successful and unsuccessful promotion strategies if they differ by channel and showcase the channels where the audience is most interested/likely to engage without work.

[1] Student equality, diversity and inclusion report 2019, University of St Andrews, <https://www.st-andrews.ac.uk/about/edi-progress-reports/student-equality-diversity-and-inclusion-report-2019> [Accessed 5 April 2024].

3.5 Discussion

a. 2023 Survey Analysis Discussion

Whilst the rent prices in St Andrews have been rising rapidly in the past few years due to increasing number of students in the same space, we hypothesise that most students choose where they live based on first available property rather than their preferences. Exploring actual rent paid, as opposed to one's willingness to pay, may have significantly reduced the explanatory power of individual characteristics. Someone may have a budget of over £1000, but if they are offered a property on Lamond Drive for £600 a month, they may take it. Similarly, we would like to test whether students' decision of where to live is randomly allocated, due to everyone simply accepting the first property offered, or whether there is a correlation between location and personal characteristics of the respondents.

We could also explore the two indicators of students' willingness to pay, through the explicit willingness to pay questions about properties on Market Street, Lamond Drive and in Dundee, and "How much would your rent have to increase for you to move?". The previous survey attempted to capture willingness to pay by presenting hypothetical properties in Dundee, on Lamond Drive, and Market Street. However, these questions were not often answered, and we suspect respondents interpreted the questions as what they expected to pay, rather than their maximum budget. Respondents for whom a family member pays for their rent (278 respondents, >50%) may also not be aware of their family member or guardian's maximum budget for rent. This is an opportunity for the new survey to gather more reflective answers.

The survey data could be of interest to all the other three teams. Modelling can utilise the willingness to pay responses to craft a demand function. Although the Register team now have the full HMO data from Fife Council, the responses could be used to confirm properties with 3 or more bedrooms. Similarly, the Price team could use the new rent data in conjunction with the previous one to examine how rent of specific properties has changed over time. (Some respondents provided us with this data explicitly, others we can match through the addresses)

These questions informed our new survey. We modified the address question to be a drop-down list of postcodes followed by a drop-down of addresses, making it easier to analyse. We added questions about how long respondent actively searched for properties before securing a contract and whether they actively considered properties outside of St Andrews.

b. 2024 Survey Roll-out

The survey promotion had two main goals: to achieve a high response rate and a representative sample. Our higher response and completion rates compared to the previous wave is an achievement and indicates that we have been successful in the first goal. Beyond the survey design the higher sample is largely down to our marketing decisions. With the notable exception of a general bias towards female participants, the survey-takers also match the overall typical characteristics of the wider St Andrews student population. To help demonstrate our professionality and engender trust, the team updated the module's style and branding. All assets created are now available for future teams in the graphics pack. We would highly recommend future teams reuse this graphic and colour scheme, having established an aesthetic which will be easily recognised by previous participants future teams are likely to benefit from this 'brand recognition.'

The survey was marketed using five core channels: St Andrews academic channels and locations (Academic presidents' newsletters and library television screens); The Students Union (emails and social accounts of SRC members); student societies; the St Andrews Housing Instagram account and word of mouth advertising. We would recommend the continued use of all these channels in further survey waves. Social media accounts appear to be the most successful source of engagement, beyond posting our own account we the survey was promoted on general St Andrews meme or discussion pages (mauricioatstandrews, fessdrews, crushdrews) and housing specific accounts (CASH and GetARoom). Our main recommendation would be to use our social media channels more effectively.

Firstly, this could be achieved by increasing engagement with the social media account all year-round, through posting project updates to generate an organic following. This was a considered outreach method yet the time commitment to results ratio was ambiguous and thus it was not carried out. Our primary suggestion for future teams would be to carry out paid online advertising, something that this year's team did not carry out. This could be useful for increasing the number of respondents as well as using targeted advertising to correct for demographic biases.

This year our survey was largely representative reflecting the expected number BAME students, international students and commuters, an oversampling of women appears to be the only characteristic that unrepresentative of the wider student population. Yet this is still concerning as it could reflect bias in unobservable characteristics, possibly undermining the validity of our survey. We would thus suggest that, possibly alongside purchasing general audience online marketing, the team should use demographic controls to increase the number of male participants.

The delay between requesting approval and posting on external social media accounts was another issue, both a longer survey time and better preparation could have likely increased our responses. Similarly, both the library graphics and inclusion in the union president were only active for less than a week before the survey closed. Other possible ignored or underutilised channels were the VIP website, the memos@st-andrews.ac.uk, and physical marketing. Though flyers were made on the request of management it appears they were never printed, or if printed, never scanned.

Following last years practice, the survey team spend money on gift card incentives. Since all our marketing materials mentioned this reward there is currently no data as to whether respondents are more likely to fill out the survey given material incentives. Data as to the importance of this incentive could be gathered by the next team. For its ability to tackle survey gender bias as well as the lack of evidence that the cash rewards provided meaningful survey engagement, we would speculatively suggest, in future, these funds may be better put towards online marketing.

c. 2024 Survey Analysis Discussion

Over the course of the project, our team has worked hard to identify relationships between *rent* and its explanatory variables to contribute to discussions of demand side factors. We had many accomplishments during our research. Using analysis we conducted on the previous survey's data we were able to identify areas of improvement in the question structure. This aided us in drafting a new survey and guided how we structures the questions from verbiage to response style (i.e. drop-down function). The new style of survey provided cleaner responses and decreased the friction for respondents, yielding a higher complete response rate. We employed a different outreach approach that contributed to our increase in responses. Instead of spending time printing flyers and handing them out in popular locations, we opted to promote the survey fully online though emails, social media channels, and personal communications. This increased our responses from 635 in four weeks to 716 responses within ten days. Additionally, using our improved survey model, we have gathered a comprehensive data on key characteristics of undergraduate and postgraduate students attending the University of St Andrews. This dataset captures a range of potential explanatory factors of rent such as fee status, ethnicity, gender. Preliminary data analysis has provided insightful information, building upon the work of the 22/23 Survey team. This will be instrumental in guiding future teams in their analysis.

Although the response data improved with the new survey structure, there is significant room for improvement. Although "how do you describe yourself" variables were not included in the regression, these data points could prove useful. Ways to diversify the responses and capture a more representative sample should be explored. The demanding timeline of the project has restricted the analysis conducted by our new on the 2024 survey data. As a result, multiple explanatory variables of *rent* were not explored; subsequent teams should endeavour to expand our analysis to examine the impacts of other response data. Additionally, limitations on the "Who is primarily responsible for paying your rent?" question responses may have created some skewed results. Allowing respondents to select multiple sources of income (i.e. student loans and own income) should be explored to accurately capture respondent behaviour.

d. Opportunities for Further Exploration

There is plenty of scope for further analysis of the data collected in the new survey. We did not have time to import the rent data for halls residents into the dataset. Once that is complete, the number of rent observations will increase by 220 and allow future teams to explore the characteristics of those who choose to remain in halls and which halls they choose to stay in.

We also collected information about the subject of the respondent, which could be analysed to find out whether subjects with higher forecasted earnings after their degree have a higher willingness to pay. This could be combined with what we know about their financial situation to see, for example, whether those with a scholarship who study high earning potential subjects pay more for their accommodation compared to those who have lower earning potential.

The willingness to pay question has not yet been explored. We asked about respondents' individual monthly rent budget, and this could be modelled alongside numerous other answers. Firstly, we could examine whether there is a disparity between what students actually pay and what they are willing to pay. This would tie into our hypothesis that distribution of properties has less to do with the characteristics of renters than would be expected, due to the highly competitive nature of the rental market in St Andrews. We could model monthly rent budget as an approximation of willingness to pay and find out whether the factors we explored for rent in general can also be applied to rent budget.

The other possible Y variables that could be examined to shed light on the housing environment include applying for properties outside of St Andrews, declining an offer of a property, and how many months respondents actively searched for somewhere to live. We also have additional X variables such as Sports Centre visitation frequency, bicycle and car ownership, and the Likert scale of factors regarding accommodation preferences that the respondents completed.

There is also scope for a time series of properties in the town – it would be very interesting if the Supply team could utilise the two datasets to see how the rents have changed over time, by matching addresses from each survey.

Conclusion

The team has made significant contributions to understanding student rent preferences and our survey design and marketing plan was instrumental in gathering this result. We have demonstrated that the financial characteristics of students are significant in explaining the demand side of the student housing market. This result, though limited by possible sample bias and the issues with our Gauss Markov assumptions already stated, is important to understanding the impacts of increasing rent on students. If fee status and funding sources continue to predict rent paid this may suggest that certain groups could be priced out of the town. However, many aspects of student preferences as well as the possibility of constructing panel data, remain unexplored.

4. Register Team

Fife Council is legally required to maintain a publicly accessible database of all active HMO licences in Fife.[2] This publication discloses information relating to licences issued within the past 5 years including the property's address, the applicant's name, the letting agent, the number of occupants, the licencing decision, and the licence issue, expiration, and application dates. An example observation from the published HMO Register PDF is given below.

App Ref Number	Applicant Name/s	WARD	HMO Address	Agent Name	License Status	Date of Ap	Date Issued	Expire Date	Tot Occs	Decision
F02896/18	Jan Halley	W5R22	Hyndhead Hostel Michael Street Buckhaven KY8 1JP		Licence Expired	27/08/2018	27/08/2019	30/10/2021	11	CON

An observation from the Fife Council HMO Public Register (Q2 2023)[3]

The register team aimed to analyse the Fife Council HMO Register to identify trends in the supply of HMO-licenced properties over time and evaluate the impact of the HMO Overprovision Policy on housing supply. During the previous academic year, the team produced various descriptive statistics on the supply of HMO-licenced properties in St. Andrews and ran a regression to identify selective licence attrition based on property characteristics. However, these results were derived from flawed data inherited from a previous team's attempt to manually convert the register from PDF into a machine-readable format. Consequently, the findings were inaccurate and produced solely as proof-of-concept, as outlined in the Martinmas Semester 2023/24 Progress Report. This year, our work has focused on obtaining accurate register data and replicating our analysis from the previous year. We successfully obtained the HMO register dating from 2010 to 2024 from the Council using Freedom of Information legislation[4]. Simultaneously, we converted the public register PDF into a machine-readable format using Stata and Python. Using both datasets, we obtained basic descriptive statistics and ran selective attrition regressions. Finally, we analysed the Short Term Let Register which is maintained by Fife Council as required under legislation[5]. This dataset provides information on currently actively short-term let licenses and applications. We also produced a range of supplementary data files to support other teams' research including a database of inflation-adjusted property values based on the most recent sale price of all properties in St Andrews derived from the Land Register of Scotland and a database of the Council Tax Bands assigned to dwellings in St Andrews using data from the Scottish Assessors Association.

The register team included Kieran Pirie (kmp21) and Finn Watson (forw1). Tasha Delvecchio (td61) was also a member of the team during the first half of the semester.

[2] The Housing (Scotland) Act 2006 Part 5
[3] Fife Council (2023) "HMO Public Register – Q2 2023"
[4] Freedom of Information (Scotland) Act 2022
[5] The Civic Government (Scotland) Act 1982 (Licensing of Short-term Lets) Order 2022

4.1 Data

This section begins with an overview of the datasets used in our analysis, followed by a discussion on data gathering using Freedom of Information legislation, and finally a presentation of key descriptive statistics.

HMO Register

Fife Council publishes periodic versions of the HMO Register online in accordance with the Housing (Scotland) Act 2006 Part 5. However, they publish the data in .pdf format which is not machine-readable and thus unsuitable for data analysis. Therefore, we submitted a Freedom of Information (FOI) request to Fife Council, as described subsequently, to request a copy of the register from 2010-present in .csv format.

We received a version of the file that was exported on 9th February 2024 so reflects the status of applications on that specific date. The dataset contains ten variables at the license level including address, ward, application reference number, licence status, application decision, occupancy, application receipt date, licence issue date, and licence expiry date. Additionally, the "LISTAT" variable contains a coded version of the license status variable. An example observation is given below.

App Ref	Ward	HMO Address	LISTAT	Status Desc	Received	Issued	Expiry	Tot Occs	Decision
F02889/18	St Andrews	4 West Port Court Bridge Street St Andrews Fife KY16 9FB	7_EXP	Licence Expired	29/06/2018	08/11/2018	25/08/2021	4	Grant Licence

An observation from "All HMO Apps.xlsx"

Several variables contained in the public register are missing from the historic register we received through FOI. This is because the Council were unable to export certain variables (such as applicant name and address) in .csv format.

We cleaned the dataset by adding missing postcodes and merging it with a supplementary dataset containing postcode-level information such as coordinates and walking distance to 100 Market Street. The process used to produce the supplementary dataset is described subsequently. We also converted the dataset to long-form, which presents an observation for each license for every month between January 2010 and February 2024 and indicates whether the property is active in any given month using the issue and expiry date variables. The cleaned license-level dataset is available here, and an example observation is given below. The Stata code used to produce the cleaned dataset is presented in Appendix 5.

AppRef	postcode...	HMOAddress	LISTAT
F04329/21	KY16 9LY	Flat 5 Forbes HouseDavid Russell ApartmentsBuchanan GardensSt AndrewsFife	5_ISS
F04329/21	KY16 9LY	Flat 5 Forbes HouseDavid Russell ApartmentsBuchanan GardensSt AndrewsFife	5_ISS

LISTAT	StatusDesc	Received	Issued	Expiry	TotOccs	Decision
5_ISS	Issued	10/7/2021	4/3/2023	11/4/2024	6	Grant with Conditions
5_ISS	Issued	10/7/2021	4/3/2023	11/4/2024	6	Grant with Conditions

Distanceto...	Xeasting	Ynorthing	Latitude	Longitude	year	month	Active
2	349273	716281	56.336164	-2.82207	2023	4	0
2	349273	716281	56.336164	-2.82207	2023	5	1

An observation from "FOIRegister_Cleaned.dta"

We also produced "FOIRegister_UniRemoved_Cleaned" which is identical in all respects except for its exclusion of university halls. We created this dataset to analyse the impact of the Overprovision Policy on the private rental sector. The Stata code used to produce the cleaned long-form data with university licences removed is outlined in Appendix 6.

Finally, we produced a property-level dataset (a derivative of the license-level dataset with university properties removed) to conduct the selective attrition regression. The process involved algorithmic matching of addresses and postcodes complimented by manual matching and review. Each license held by a property is numbered from 1 to 8, and corresponding variables are created to capture license-level information. For example, in the observation presented below, license 4 (AppRef_4) was applied for on 2 June 2010 (Received_4), granted with conditions (Decision_4) on 6 October 2010 (Issued_4), and expired (StatusDesc_4/LISTAT_4) on 7 June 2013 (Expiry_4). The numbering of the licenses is not chronological. The Stata code used to produce the property-level dataset is outlined in Appendix 7.

id_failed...	HMOAddress	Postcode	Datamatc...	Ward	TotOccs	Description	GridReference
2	103 Bridge Street St Andrews KY16 8AA	KY16 8...	8	St Andrews	5	KY16 8AA Bridge Street	NO 50634'1...

Xeasting	Ynorthing	Latitude	Longitude	check	id	AppRef_0	LISTAT...	StatusDesc_0	Received_0	Issued_0	Expiry_0	Decision_0
350634	716238	56.335922	-2.8000511	.	232	F01865/16	7_EXP	Licence Expired	4/11/2016	9/29/2016	6/7/2019	Grant with Conditions

LISTAT...	LISTAT...	LISTAT...	LISTAT...	LISTAT...	LISTAT...	LISTAT...	StatusDesc_1	StatusDesc_2	StatusDesc_3	StatusDesc_4	StatusDesc_5	StatusDesc_6	StatusDesc_7
7_EXP	5_ISS	7_EXP	7_EXP	7_EXP			Licence Expired	Issued	Licence Expired	Licence Expired	Licence Expired		

Decision_1	Decision_2	Decision_3	Decision_4	Decision_5	Decision_6	Decision_7
Grant with Conditions	Grant with Conditions		Grant Licence	Grant Licence		

Received...	Received...	Received...	Received...	Received...	Received...	Received...	Issued_1	Issued_2	Issued_3	Issued_4	Issued_5	Issued_6	Issued_7
13may2019	17jun2021	07jun2006	02jun2010	22may2013	.	.	13apr2020	16dec2022	22nov2007	06oct2010	23jan2014	.	.

Expiry_1	Expiry_2	Expiry_3	Expiry_4	Expiry_5	Expiry_6	Expiry_7
07jun2022	18dec2025	07jun2010	07jun2013	07jun2016	.	.

An observation from "FOI_Register_PropLevel_Final.dta"

Several variations of the license-level dataset were produced for other teams:

- "ActiveHMObyNumberofProps_byPostcode_2012to2024.xlsx" – presents the number of active licenses per postcode per year and the ratio of active licenses to the number of properties in each postcode by year. The number of properties in each postcode by year was calculated using a supplementary dataset obtained from the Scottish Assessors Association, which we discuss later. This data was shared with the demand team for inclusion in their hedonic pricing regression and with the supply team to quantify the supply of HMO properties in their model. In the future, this dataset may be used for a dedicated study of localised studentification. The*.do file used to produce this file is available here.
- "MonthlyActiveBedrooms.xlsx" – presents the number of licensed occupants by month. We also produced smoothed numbers using a twelfth-order moving average. This data was shared with the supply team for inclusion in their model. The*.do file used to produce this file is available here.
- "MonthlyActiveLicenses.xlsx" – presents the number of licensed HMO properties by month. We also produced smoothed numbers using a twelfth order moving average. This data was shared with supply for inclusion in the model. The*.do file used to produce this file is available here.

We emphasised to each team that the supply numbers after February 2024 are misleading as they do not consider the possibility of license renewals. Therefore, future teams should exercise extreme caution when interpreting the supply statistics after the date at which the file was exported by Fife Council.

HMO Register – PDF Conversion

Prior to receiving the HMO Register from Fife Council in *.csv format, we systematically converted the published .pdf version into a *.dta format using Stata. The descriptive statistics and results discussed subsequently use the files outlined above. However, upon comparing the .csv file with the dataset produced through our PDF conversion process, we identified very few discrepancies. Therefore, the PDF conversion code may be useful for future teams if they wish to analyse public registers published in .pdf format only.

When the PDF data is loaded into Stata, licence data is not formatted into discrete information variables (e.g. application reference number, applicant name, issue date). Instead, each licence 'row' is treated as a single string that contains several pieces of licence information. An example of how a licence is formatted when imported into Stata is given below.

Row 1	F02892/18	Mrs Caryn Beaton	W5R18	4 Cammo Lodge St	Licence Expired	04/07/2018	26/09/2018	23/09/2021	4	GRANT
			Row 2	Andrews KY16 9HW						
Row 3		Mr Christopher Beaton	W5R18	4 Cammo Lodge St	Licence Expired	04/07/2018	26/09/2018	23/09/2021	4	GRANT
			Row 4	Andrews KY16 9HW						

Observation from HMO Register

F02892/18 Mrs Caryn Beaton W5R18 4 Cammo Lodge St Licence Expired 04/07/2018 26/09/2018 23/09/2021 4 GRANT	Row 1
Andrews KY16 9HW	Row 2
Mr Christopher Beaton W5R18 4 Cammo Lodge St Licence Expired 04/07/2018 26/09/2018 23/09/2021 4 GRANT	Row 3
Andrews KY16 9HW	Row 4

The same observation once imported into Stata

Each row contains several pieces of discrete licence information. For example, the first row contains the application reference number, application name, ward, address, licence status, application date, issue date, expiration date, occupancy, and licence decision. Because of this, regex commands must be used to extract the relevant substring based of known patterns in the data. For example, if a licence row contains three dates (DD/MM/YYYY), the three dates will be extracted into application date, issue date, and expiration date variables, respectively. In ExpeditedPDFData_FW.do, application reference number, application date, issue date, expiration date, total occupancy, and ward information are extracted into separate variables and converted into long-form formatting to produce ExpeditedPDFData_LongFormRevised.dta (Appendix 8). An example observation is given below.

AppRefNu...	AppDate	IssueDate	ExpireDate	Andrews	firstRow	WARD	totOccs	IssueDat...	ExpireDat...	month	year	ym	ym_date	Active
F02833/18	26/04/2018	26/04/2019	29/06/2021	1	1	W5R18	4	21665	22460	4	2019	711	21640	0
F02833/18	26/04/2018	26/04/2019	29/06/2021	1	1	W5R18	4	21665	22460	5	2019	712	21670	1
F02833/18	26/04/2018	26/04/2019	29/06/2021	1	1	W5R18	4	21665	22460	6	2019	713	21701	1
F02833/18	26/04/2018	26/04/2019	29/06/2021	1	1	W5R18	4	21665	22460	7	2019	714	21731	1

An observation from "ExpeditedPDFData_LongFormRevised.dta"

Short-Term Let Register

Fife Council introduced a short-term let licensing scheme on 1st October 2022 that requires that properties used for short-term letting be licensed. Properties that operated as short-term lets before 1st October 2022 had until 1st October 2023 to apply for a licence. After that date, they could only continue to operate if they had applied for a licence before the deadline which was pending or been granted a short-term let licence. The Council indicated that it would take 12 months to determine applications submitted by existing operators. Therefore, at the time of our analysis, many applications were showing as pending whilst operating as short-term lets. Therefore, we considered all properties on the register as active short-term rental properties.

The Council maintains a public register of short-term let applications online in accordance with The Civic Government (Scotland) Act 1982 (Licensing of Short-term Lets) Order 2022. The version we analysed was published on 12 March 2024.

An example observation is presented below.

Address	Town	County	UK Postcode	UPRN	Ward	Applicaiton Type	Date Application Received	Licence Expiry Date
12a Market Street	St Andrews	Fife	KY16 9NS	320329743	19	Full	11/28/2023	Pending

Application Status	Type of Premises	Short Term Let Type	Maximum Occupancy	Year	EPC Rating	Licence Number	Number of Bedrooms
Pending	Flatted Dwelling	Secondary Letting	2	2023	C		

EPC Rating	Licence Number	Number of Bedrooms	Manager Forename	Manager Surname	Company	Address	Town	Postcode	County
C			Neil	Jarvie	Latitude 55 Ltd	6 Johnston Court	St Andrews	KY16 9PY	Fife

An observation from "STL-Public-Register-12.03.24.xlsx"

<u>Supplementary Dataset – Council Tax Valuation List</u>

We purchased a copy of the Council Tax Valuation List from the Scottish Assessors Association for £50 in February 2024. The body is responsible for maintaining a register of all heritable properties in Scotland for local taxation purposes. Summarily, the database contains information relating to every property liable for council tax in Scotland. The variable descriptors are presented in the table below.

#	Field	Description
1	ASSESSOR	Assessor area
2	UNITARY_AUTHORITY	Local Authority Name
3	UARN	Unique Assessor's Reference Number (only unique within each assessor area)
4	SAON	Address of property (structured using the BS7666 standard address format) •Secondary Addressable Object Name • Primary Addressable Object Name • Street • Locality • Town • Administrative Area • Post Town
5	PAON	
6	STREET	
7	LOCALITY	
8	TOWN	
9	ADMIN_AREA	
10	POST_TOWN	
11	PCOUT	Outward postcode (first part of postcode)
12	PCIN	Inward postcode (second part of postcode)
13	BAND	The valuation band allocated to the property
14	EFFDATE	The effective date of the Council Tax band
15	GARAGE	Domestic Garage (G) or Store (S) entered in Council Tax list as pertinent to a dwelling

An example observation is given below.

assessor	unitary_authority	uarn	saon	paon	street	locality	town	admin_area	post_town	pcout	pcin	band	effdate	garage
Fife Council	Fife	203179701		143 MARKET STREET	ST ANDREWS			FIFE		KY16	9PF	F	21/04/1...	

An observation from "SAADatabase.csv"

This dataset was originally purchased to integrate an address lookup function into the Qualtrics survey to ensure standardised address formatting. We supported the Survey team by cleaning the database and reformatting the data into a format accepted by Qualtrics. The files shared with the Survey team are available here, and the*.do file used for cleaning and reformatting is available here.

Supplementary Dataset – Postcode-Level Information

We produced a dataset containing geographic information at the postcode level, including walking distance to 100 Market Street (a point that roughly approximates the 'amenity centre' of St Andrews) according to Google Maps and geographic coordinates. The walking distance information was gathered manually in Martinmas 2021/22 and the geographic coordinates were obtained from an online batch address to coordinate convertor. This data has been used by all teams to supplement existing datasets including survey responses, price indexes, and licensed property registers.

Description	Postcode	Distance to Town	Grid Reference	X (easting)	Y (northing)	Latitude	Longitude
KY16 8AA Bridge Street	KY16 8AA	0.8	NO 50634 16238	350634	716238	56.335922	-2.8000511

An observation from "Postcode Continuous 1.xlsx"

Supplementary Dataset – Property Sale Price Database

We created a database containing the last purchase price and date for every property in St Andrews by manually scraping the Land Register of Scotland (ScotLIS) maintained by Registers of Scotland. We searched each St Andrews postcode individually and copied the output table into a spreadsheet. The data was scraped on 14 March 2024 and provided to the supply team for use in their hedonic pricing regression.

We approximated a current valuation for all properties (AdjPrice_HPI) using the Scottish Housing Price Index. We assumed that any property bought in a given year would receive the full benefit of that year's annual property value growth as reported in December of that year. However, the growth in St Andrews property valuations has exceeded the Scottish average for a significant period, so this approximation is likely downward biased. Additionally, future participants should be mindful that the last sale price does not necessarily reflect current value if property owners have made improvements/renovations.

Since property titles can encompass multiple physical properties, we added a caution variable (Caution_MultipleProperties) that equals 1 when another property within the same postcode has been sold for the same price on the same day (indicating the price listed is for a multi-property title and should not be considered). Additionally, properties are occasionally sold for extremely low sums (e.g., £1) for inheritance tax planning purposes. Therefore, we added a caution variable (Caution_MisleadingPrice) that equals 1 when AdjPrice_HPI < 20,000. The*.do file used for cleaning is available here.

An example observation is presented below.

Address	Date	Price	Postcode	Year_Sold	AdjPrice_HPI	Caution_MultipleProperties	Caution_MisleadingPrice
2 ST. MARY STREET	4/11/2008	610000	KY168AY	2008	923596	0	0

An observation from "ScotLIS_Clean_Mar2024.xlsx"

Freedom of Information Data Gathering

Under British law, the public has the right to request any information held by public authorities including local authorities and universities[6]. This semester, we embarked on a large-scale data-gathering campaign using Freedom of Information (FOI) to request data from 11 local authorities, 15 universities, and 66 individual Colleges at the universities of Cambridge and Oxford. The responses and a selection of FOI request templates are available here.

The details of our requests are outlined below:

- We requested a copy of the HMO register dating from 2010 to the present from each local authority.
- We also requested information on rejected HMO license applications, reports of unlicensed HMO properties, objections to HMO licence applications, and traffic survey data from Fife Council.
- We requested information on accommodation rent prices, student enrolment, and student hall occupancy from each university.

We cleaned and formatted the road traffic survey data provided by Fife Council, as the .txt files provided contained multiple traffic surveys that needed to be split into individual .dta files. We also converted the traffic surveys into long data format. After cleaning, we counted 1,922 traffic survey data files. The *.do file used for the data cleaning is available here, and the cleaned traffic survey files are saved here. A selection of observations from a cleaned traffic survey data file is presented below.

[6] Under the Freedom of Information Act 2000 in England and the Freedom of Information (Scotland) Act 2002 in Scotland

	date	day	time	street	traffic_count	date_stata	direction
31	2 March 2013	Sat	16:00	A915 St Andrews, Bridge Street	906	19419	Total Flow
32	2 March 2013	Sat	17:00	A915 St Andrews, Bridge Street	757	19419	Total Flow
33	2 March 2013	Sat	18:00	A915 St Andrews, Bridge Street	558	19419	Total Flow
34	2 March 2013	Sat	19:00	A915 St Andrews, Bridge Street	441	19419	Total Flow
35	2 March 2013	Sat	20:00	A915 St Andrews, Bridge Street	304	19419	Total Flow
36	2 March 2013	Sat	21:00	A915 St Andrews, Bridge Street	241	19419	Total Flow

A selection of observations from "A915StAndrews,BridgeStreet_March2013_TotalFlow.dta"

The exact use case for each dataset has yet to be determined, but future research avenues may include comparing the housing market in St Andrews to other similarly sized university towns, such as Durham. Additionally, the comprehensive traffic survey data could be used for modelling the emissions impact of commuting students.

Data Organisation Flowchart

The flowchart provided overleaf delineates the ancestry and relationships among all pertinent register outputs and data files documented in this report. The chart begins by describing the data files' origin, distinguishing between those developed by our team and acquired from external sources. Green cylinders represent these data files. The red rectangles indicate the outputs featured in this progress report, along with the data file they originated from. The analyses are illustrated as circles, differentiating them as descriptive statistics, or data cleaning.

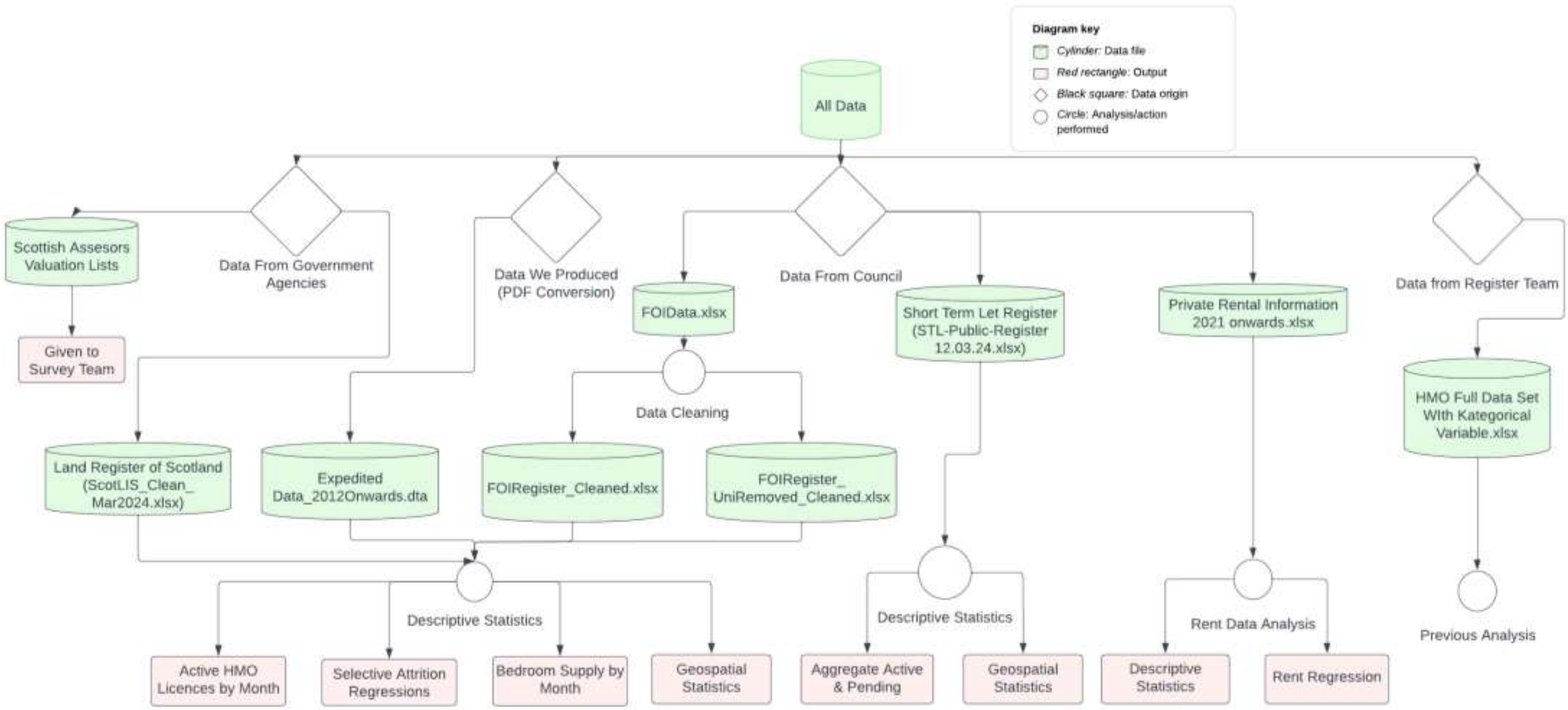


Diagram key
Cylinder: Data file
Red rectangle: Output
Black square: Data origin
Circle: Analysis/action performed
All Data
Data From Government Agencies
Data We Produced (PDF Conversion)
Data From Council
Data from Register Team
Scottish Assesors Valuation Lists
Given to Survey Team
FOIData.xlsx
Data Cleaning
Short Term Let Register (STL-Public-Register 12.03.24.xlsx)
Private Rental Information 2021 onwards.xlsx
HMO Full Data Set WIth Kategorical Variable.xlsx
Land Register of Scotland (ScotLIS_Clean_ Mar2024.xlsx)
Expedited Data_2012Onwards.dta
FOIRegister_Cleaned.xlsx
FOIRegister_ UniRemoved_Cleaned.xlsx
Descriptive Statistics
Descriptive Statistics
Rent Data Analysis
Previous Analysis
Active HMO Licences by Month
Selective Attrition Regressions
Bedroom Supply by Month
Geospatial Statistics
Aggregate Active & Pending
Geospatial Statistics
Descriptive Statistics
Rent Regression

HMO Supply Descriptive Statistics

Using the FOI HMO register data from Fife Council we generated descriptive statistics to illustrate trends in the supply of HMO-licenced properties and available bedrooms over time.

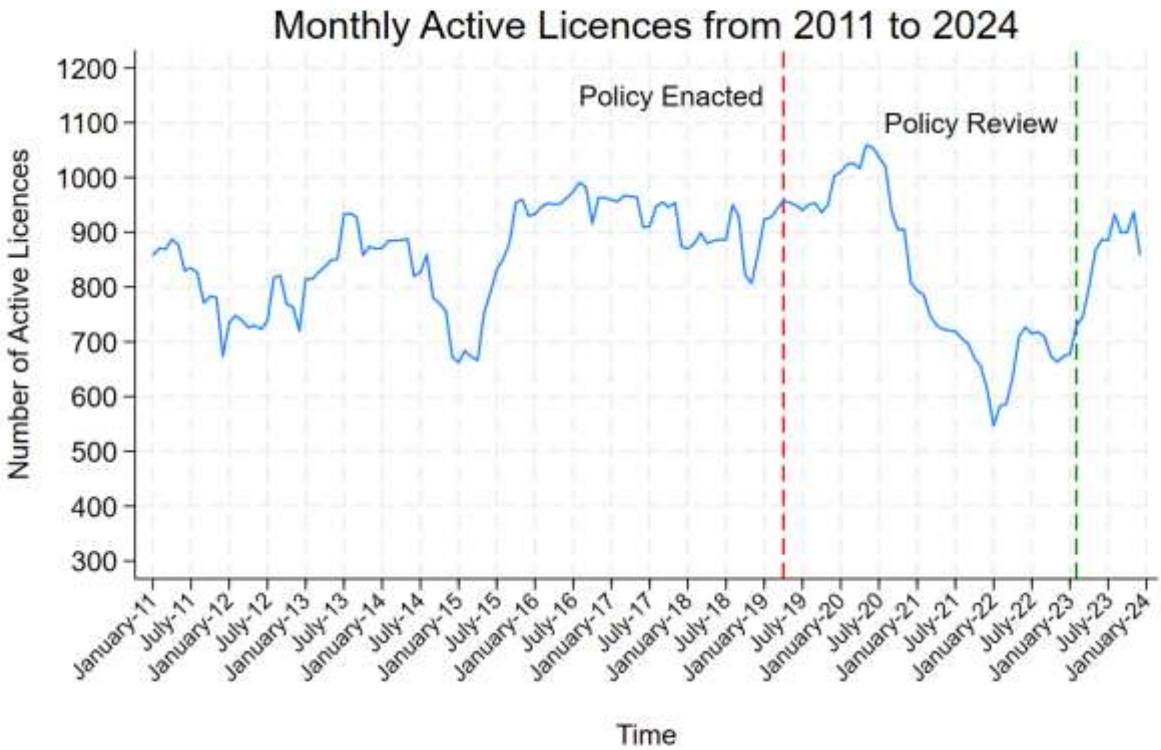


Figure 24: Active Licenses by Month (Stata code in Appendix 9)

First, we produced a graph of the number of active licences by month from the beginning of 2011 to the end of 2023. Across the entire dataset, the number of active licenses ranged from 547 to 1059.

Before the introduction of the HMO Overprovision Policy, from January 2011 to August 2019, the number of active HMO licences ranged from 662 to 991, occurring in January 2015 and July 2016, respectively. At the start of the dataset, there were 857 active licences which grew to 951 active licenses immediately before the implementation of the Overprovision Policy. While there was a significant drop in active licences in April 2015 to a total of 662 licences and high volatility during the period, the number of active licences generally increased during this period.

After the implementation of the policy in August 2019, the number of active licences continued to increase to a peak of 1059, which occurred in May 2020, after which a precipitous decline occurred throughout the pandemic period until January 2022 when total active licences numbered only 547. After the nadir of supply, active licence supply expanded, reaching a post-policy peak of 937 in December 2023. The post-policy peak in 2020 to the local peak in 2023 represents a marked decline in the number of active licences of 11.52%. The average number of active licences was 949 in 2019, compared to 843 in 2023, indicating a decline in the steady-state supply of HMO licences after the implementation of the Overprovision Policy.

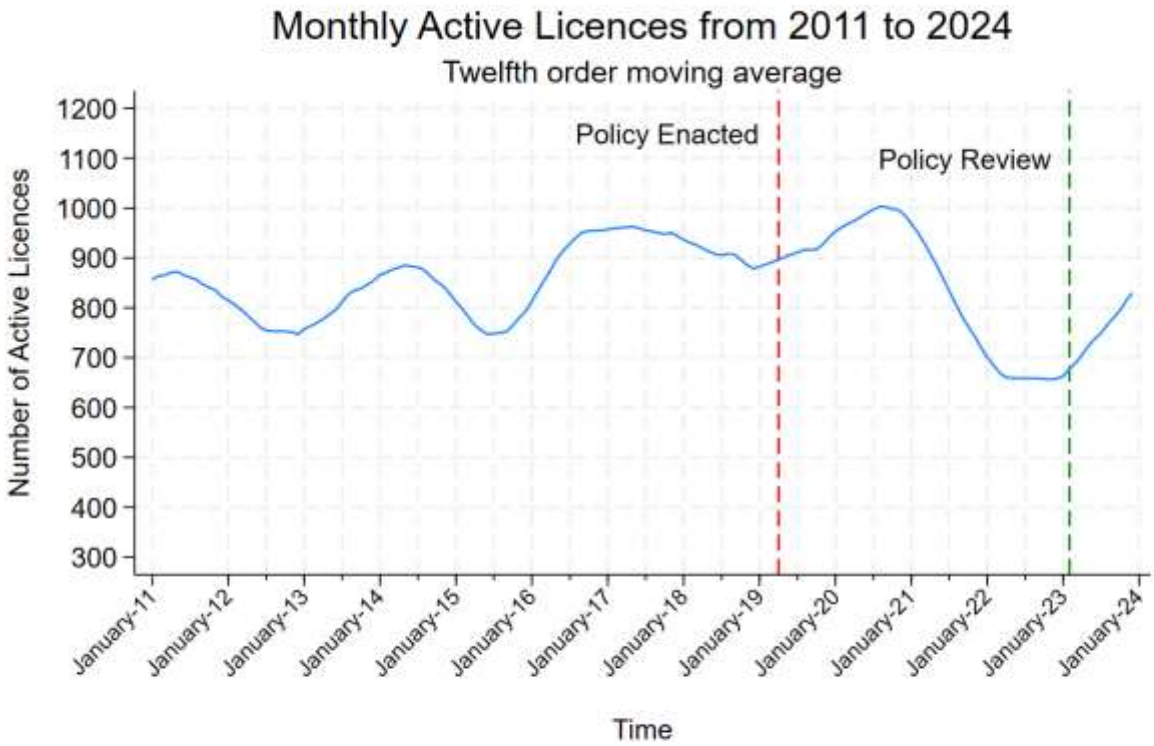


Figure 25 Moving Average of Active Licences by Month (Stata code in Appendix 10)

Figure 25 illustrates the change in active licences but uses a twelfth-order moving average to smooth supply by filtering out noise from fluctuations in the preceding 12 months. This figure shows similar trends in aggregate occupancy statistics. In the period preceding the policy, the moving average occupancy ranged between 746 and 962 with a mean of 858. In the period following the policy, the moving average number of active licences ranged from 656 to 1003 with an average of 819. The decline in licencing from the peak in 2020 to the end of the dataset was larger in the moving average data at 17.25%.

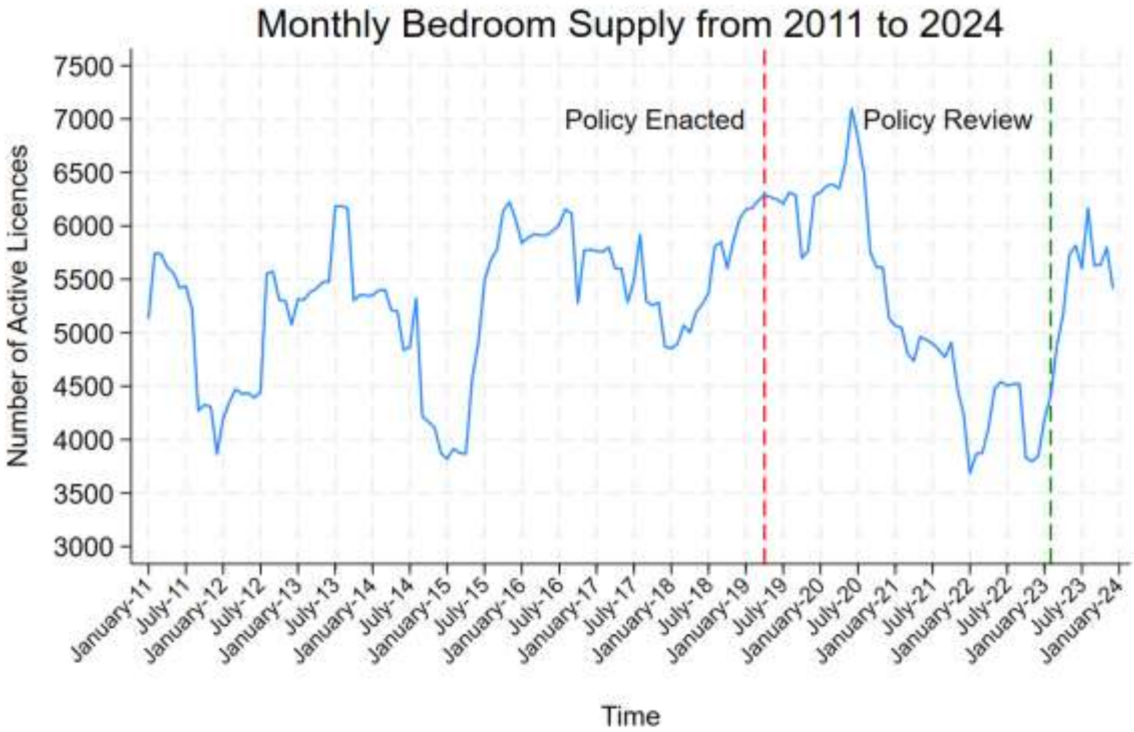


Figure 26: Monthly Bedroom Supply (Stata code in Appendix 11)

Figures 26 and 27 illustrate changes in the number of total occupants across all licenced properties in St. Andrews. Analysing student housing supply in terms of active occupants instead of the number of active licences more effectively describes the state of housing supply. Across the entire period, total occupancy in licenced properties ranged from 3,684 to 7,096 in January 2022 and June 2020, respectively. Similarly to figures 24 and 25, there was a general upward trend in the period preceding the adoption of the Overprovision Policy, with occupancy starting at 5,132 and ending at 6,312. Additionally, there was continued growth in total occupancy between the announcement of the policy and June 2020, when occupancy reached 7,096 before dropping precipitously 3,684 in 2022. After the pandemic, licenced occupancy recovered to a local peak of 6,167. From the peak in 2020 to the local peak in August 2023 total occupancy dropped 13.09%, a proportionally larger drop than in figure 25 which suggests that properties with more occupants were more likely to become unlicenced than smaller properties after the policy came into effect.

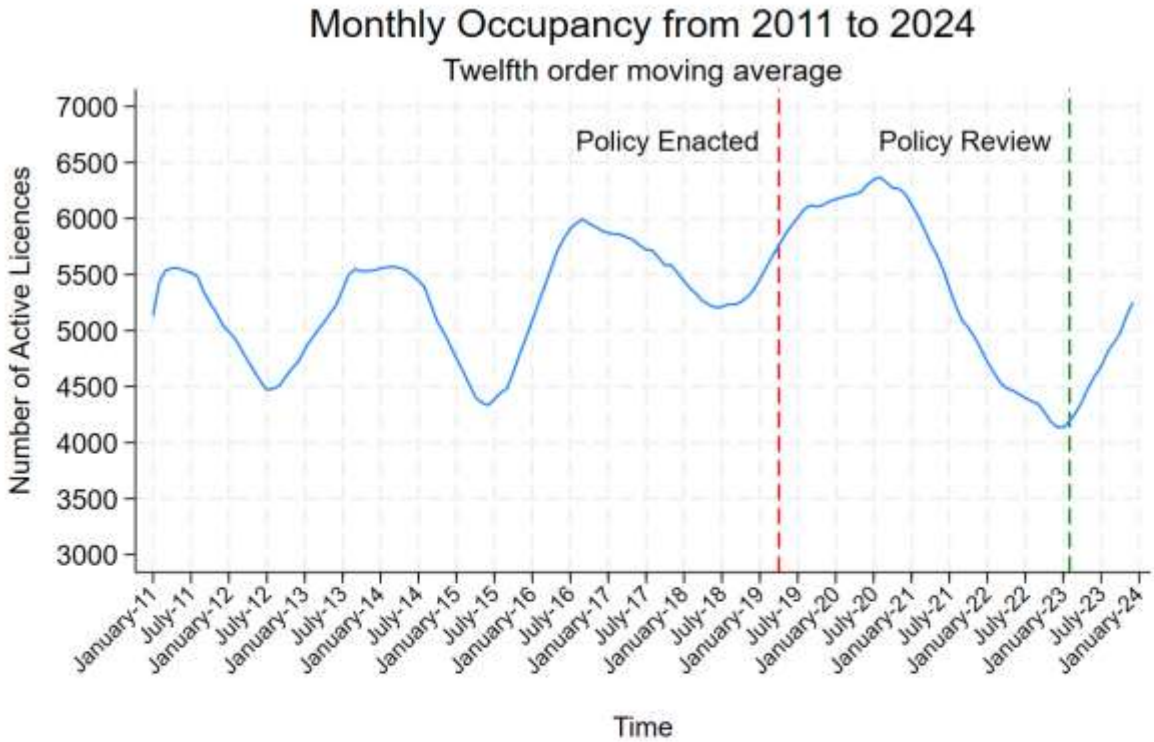


Figure 27: Monthly Moving Average of Total Bedroom Supply (Stata code in Appendix 12)

Similarly to figure 24, figure 27 uses a twelfth order moving average to show supply trends while smoothing local variation by taking the average of a given months and the 11 months preceding it. From 2011 to the end of 2023 the smoothed occupancy ranged from 4,136 to 6,368 with an average of 5,280. After the Overprovision Policy went into effect in 2019, smoothed occupancy maintained the same range but averaged higher than the total period at 5326. The smoothed post-policy decline in licencing, from a peak of 6,368 to the nadir of 4,136, represents a 35% decline.

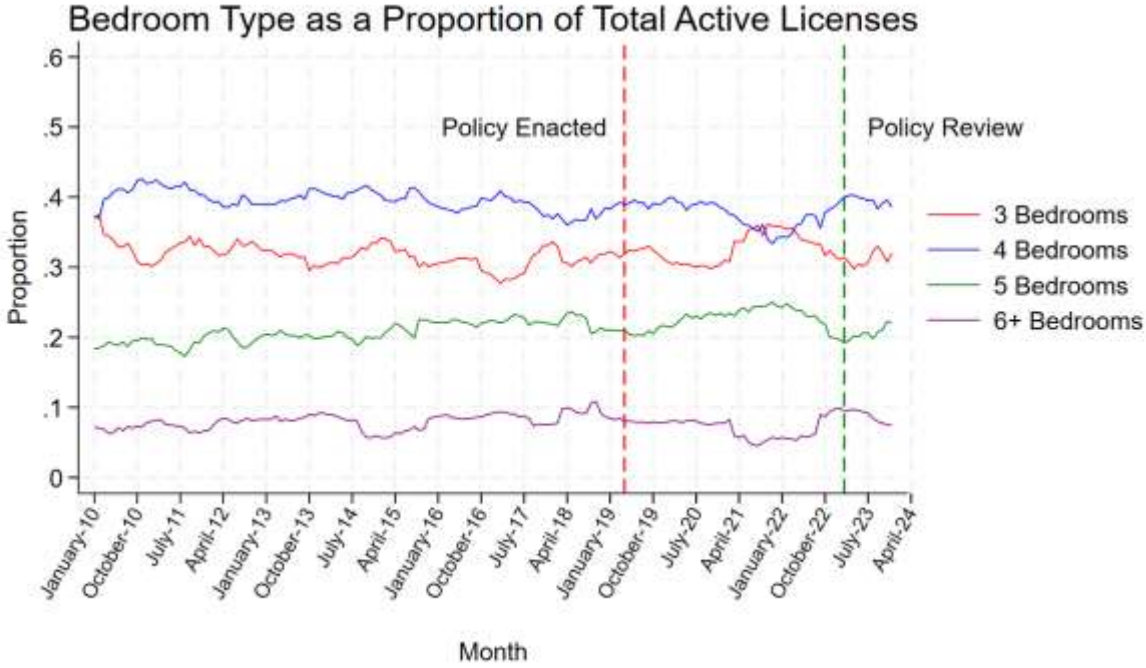


Figure 28: Bedroom Type as a Proportion of Total Active Licenses by Month (Stata code in Appendix 13)

Figure 28 illustrates how the proportional breakdown of licensed occupancy evolves over time. From 2010 to 2024, we observe a broadly consistent trend with four-bedroom properties comprising most of the licensed HMO housing stock (c. 40%) followed by three beds (c. 30-35%), five beds (c. 20%), and 6+ beds (< 10%). This graph excludes university halls of residence.

The only notable deviation from the trend occurs between April 2021 and October 2022, when the relative proportions of 4-beds and 6-beds decrease by around 5%, and the relative proportions of 3-beds and 5-beds increase by the same amount. For a brief period in mid-2021, 3-bed properties overtake 4-bed properties to become the largest proportion of active HMO properties.

Geospatial Descriptive Statistics

Using QGIS, we plotted a variety of maps to show geospatial trends in housing supply.

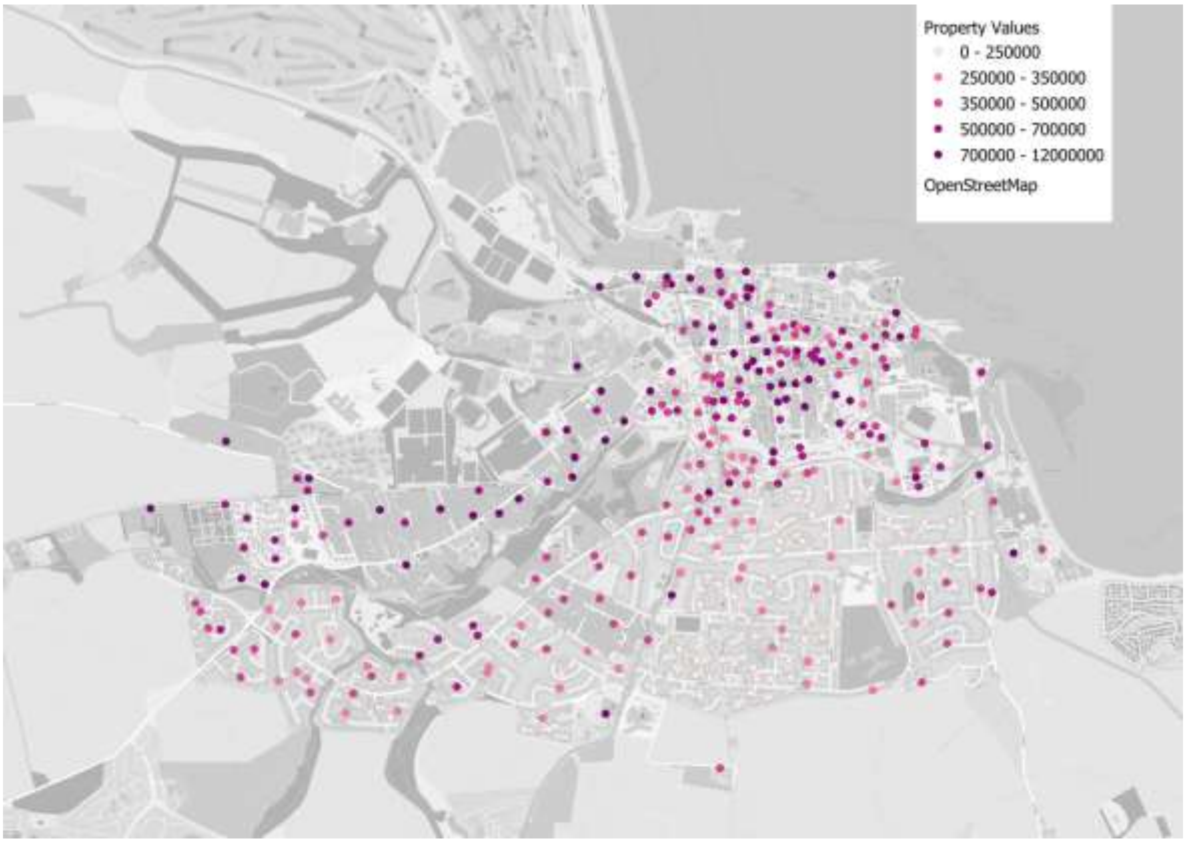


Figure 29: Colour graduated dot map of average property values by postcode

Figure 29 plots the average property value by postcode using the HPI-adjusted current value variable contained within the "Property Sale Price" dataset. We observe a clustering of high-value properties within the conservation area and along Hepburn Gardens. Contrastingly, we observe the most affordable properties in the area south of Lamond Drive and west of Greyfriars RC Primary School.

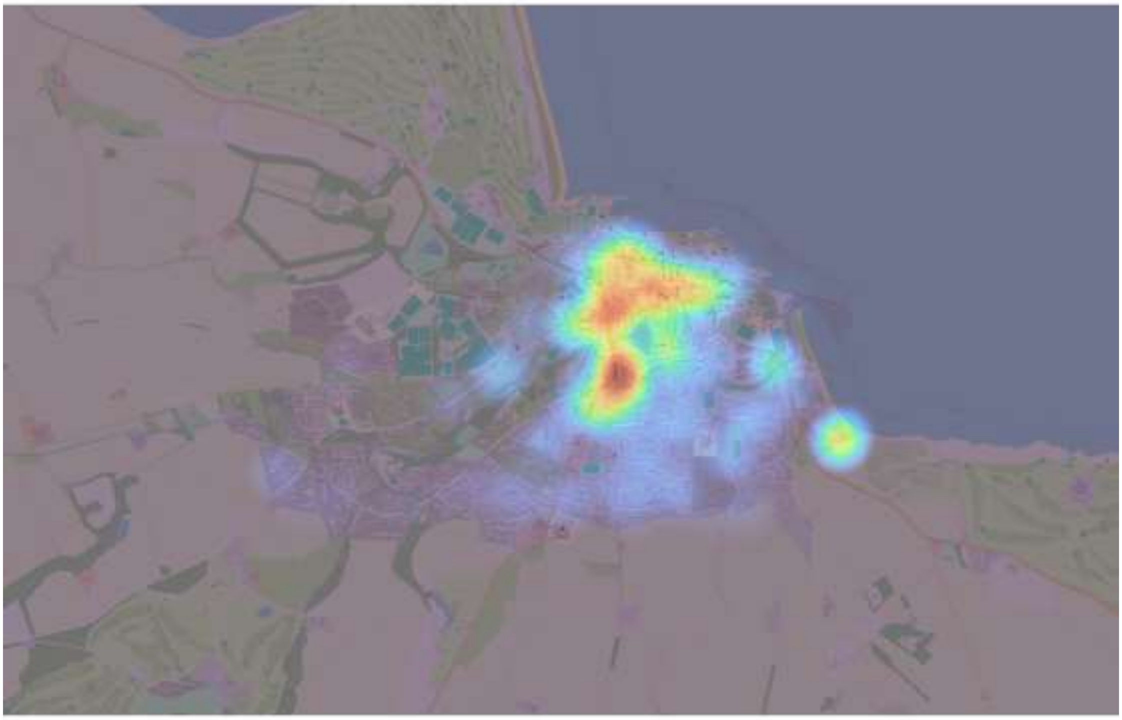


Figure 30: Heatmap of active HMOs in January 2024 (weighted by occupancy)

Figure 30 plots a heatmap of active HMO occupancy as of January 2024 and indicates significant clustering around the centre of town and wider conservation area. Interestingly, the map indicates that the area surrounding Nelson Street and the west of Kinnessburn Road has the highest concentration of active HMOs. The hotspot in the east of the town is due to East Shore, a private student hall of residence. Relating back to Figure 29, the heatmap suggests that the majority of HMO accommodation is located in postcodes with average to above-average property values.

Figure 31: Point Cluster Map of 'Lost' HMO Properties Between 2018 and 2023

Finally, we plotted a clustered dot density map of HMO properties that held an active license in 2018 but did not in 2023, as shown in Figure 31. The spatial distribution of 'lost licenses' does not seem overly dissimilar to the distribution of active HMOs in January 2024 as shown in Figure 30. This may suggest the absence of systematic license attrition based on location, although we will discuss this matter further in the results section.

Short-Term Let Register Descriptive Statistics

We briefly analysed the short-term let register to gain a sense of the scale of short-term letting in St Andrews. As discussed previously, we considered pending and granted license applications to be active due to the recent introduction of the licensing scheme and Fife Council's acknowledgement that existing short-term let properties with a pending application may continue to operate.

We identified 558 properties on the short-term let register within St Andrews, of which 480 (86%) had applied for a secondary letting license – the type of license required for letting a property in which the owner does not usually reside. The remaining applications were for home-sharing and home-letting licenses. This corresponds to a total short-term let occupancy of 3,192 persons. For context, the town's short-term let occupancy is 23% larger than the number of occupants living in licensed HMO properties in February 2023, according to Fife Council, who calculated 2,598 licensed occupants in HMO accommodation by tabulating active and pending licenses (excluding university halls). Based on Fife Council's claim in the Overprovision Policy review meeting agenda that there are 6,861 dwellings in St Andrews, short term lets represent 8.1% of the housing stock in the area.

We attempted to match properties that appeared on the HMO register and short-term let register and identified 49 properties with a combined occupancy of 326 persons. Of these, 30 properties had a currently active HMO license, corresponding to an aggregate occupancy of 210 persons. Based on the rudimentary matching process used, this number may be an understatement and should be validated by future teams.

While this may indicate that some landlords are retaining a scarce HMO license but, in fact, renting their property short-term, we are currently unable to determine whether the properties are used for both purposes seasonally (i.e., rented as an HMO during term time and as a short-term let over the summer). Future teams may wish to consider investigating each individual property to identify its use during university term time. As proof of concept, we searched Google for the addresses of several matched properties and found some available for short-term rental year-round.

The *.do file used for the short-term let analysis is available here.

Rent Descriptive Statistics

The team conducted an analysis on another dataset that was provided by Fife Council in late November 2023 while waiting for the FOI request response. The dataset named "Private Rental Information 2021 Onwards.xlsx" includes properties in Fife rented over three academic years (2021-2022 to 2023-2024) with information on the property type, rent per calendar month, location, number of bedrooms, and year the property was rented. The primary goal was to understand how these factors influence rental prices in Fife. An example observation is given below.

Year	Quarter	NoofPrope...	Advertised	Address	Town	County	Country	LHSArea	HMA	PropertyType	NoofBedro...	FurnishedUn...	RentPCM	LettingAgent
2021-22	Q1	1	www.zoopla.c...	Sunnybraes Terrace	Steelend	Fife	Scotland	West Fife	Dunfermline & West Fife	Flat	2	Unfurnished	450	Fife Letting Service Ltd

An observation from "Private Rental Information 2021 onwards.xlsx"

In order to produce descriptive statistics, the data first had to be cleaned. There were typographical errors that needed to be fixed to ensure consistency across all the observations. The main variable of interest was *RentPCM*, but when looking at how rent varies with the number of bedrooms, rent per person was considered a more appropriate measure since it is likely that the more bedrooms there are, the higher the rent will be. The dataset did not contain any information on the rent per person. To allow for an estimation of the rent per person, a new variable, *RentPP*, was generated based on the assumption that each bedroom is occupied by one person and hence calculated by dividing the variable *RentPCM* by *NoofBedrooms*. One outlier was removed from the data set because it was not an accurate representation of the rental properties in Fife. The observation was a large bed and breakfast property also available for event rental with *RentPCM* at £12000. The Stata code used to clean the rent data is outlined in Appendix 14

Once the Rent data from Fife Council had been prepared, we conducted our analysis by generating descriptive statistics to depict how rent can be affected by different factors. The team first started by graphing the distribution of *RentPCM* as well as *RentPP*.

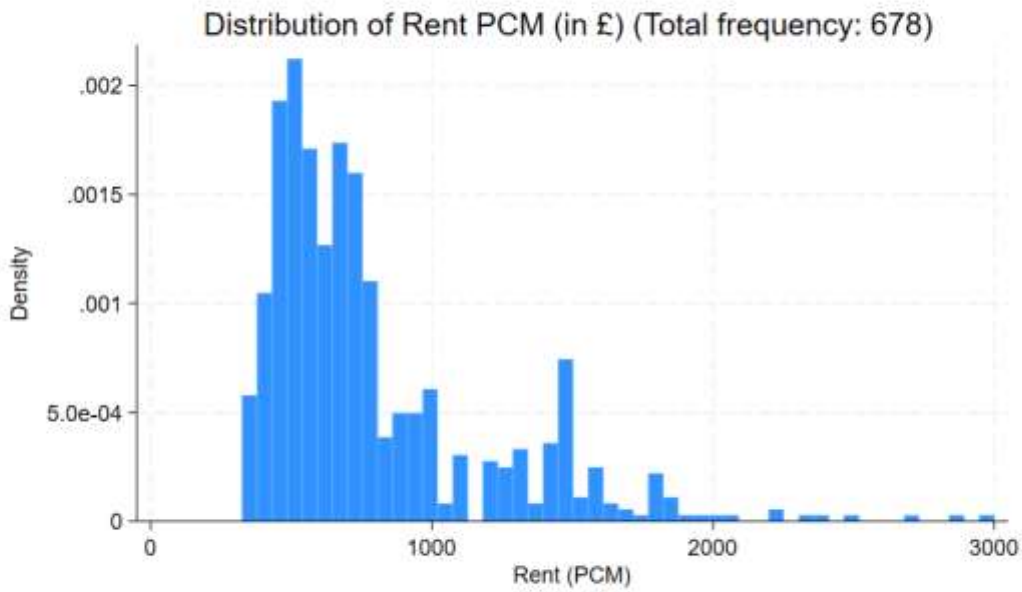


Figure 32: Distribution of Rent Per Calendar Month (Stata code in Appendix 14)

Figure 32 illustrates the distribution of the variable *RentPCM*. The values range from £325 per month to £3000 per month with a mean of £792.71. The distribution is right skewed indicating that most of the properties are clustered at the lower end of the rent range, and the mean is larger than the median, which is £650.

Figure 33 shows the distribution of the variable *RentPP* which ranges from £123 per month to £1997 per month with a mean of £394.98. The distribution is more positively skewed than Figure 31, where most of the observations lie at the lower end of the rent per person range. There are very few observations with significantly higher rent per person.

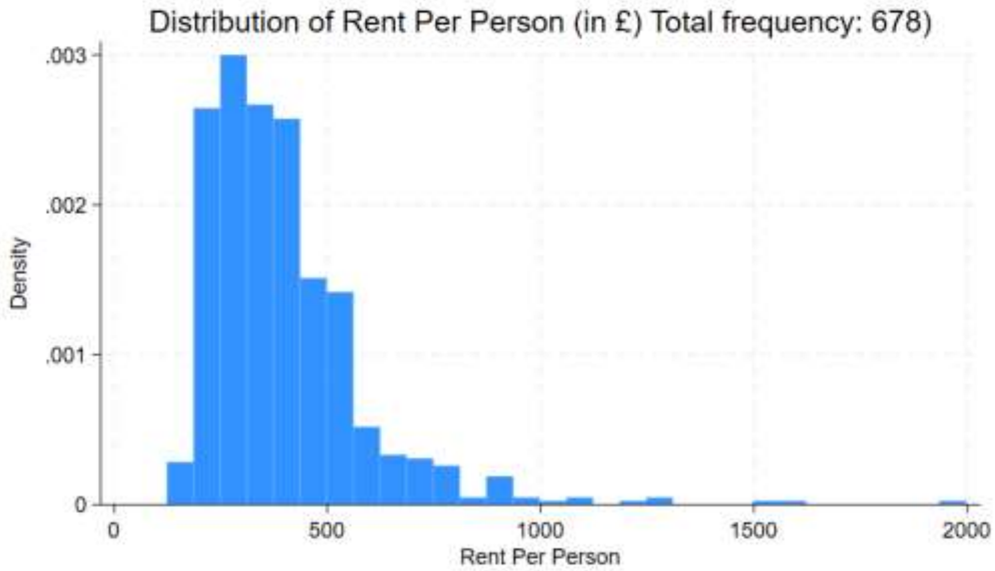


Figure 33: Distribution of Rent Per Person (Per Calendar Month) (Stata code in Appendix 14)

After graphing the distributions of the main variables, *RentPCM* and *RentPP*, the team generated cross-distributions of rent per calendar month and rent per person with the other variables to identify any underlying relationships between variables, comparing categories, highlighting disparities and exploring the different effects of the variables on rent.

Figure 44: Histograms illustrating Rent Per Person by Occupancy (Stata code in Appendix 14)

In Figure 44, there is a collection of histograms to compare the distribution of the variable *RentPP* by *NoofBedrooms*. Looking at the frequency in each histogram, there are very few observations in this dataset that are 0-bedroom (studio) and 5-bedroom properties. The majority of properties have 1, 2, or 3 bedrooms. From the distributions above, one can see that the rent per person decreases when the number of bedrooms increases from 0 to 3 and from 4 to 5. However, when the property changes from a 3-bedroom to a 4-bedroom, the rent per person increases. The reason behind the increase in rent per person at this specific interval is still unknown.

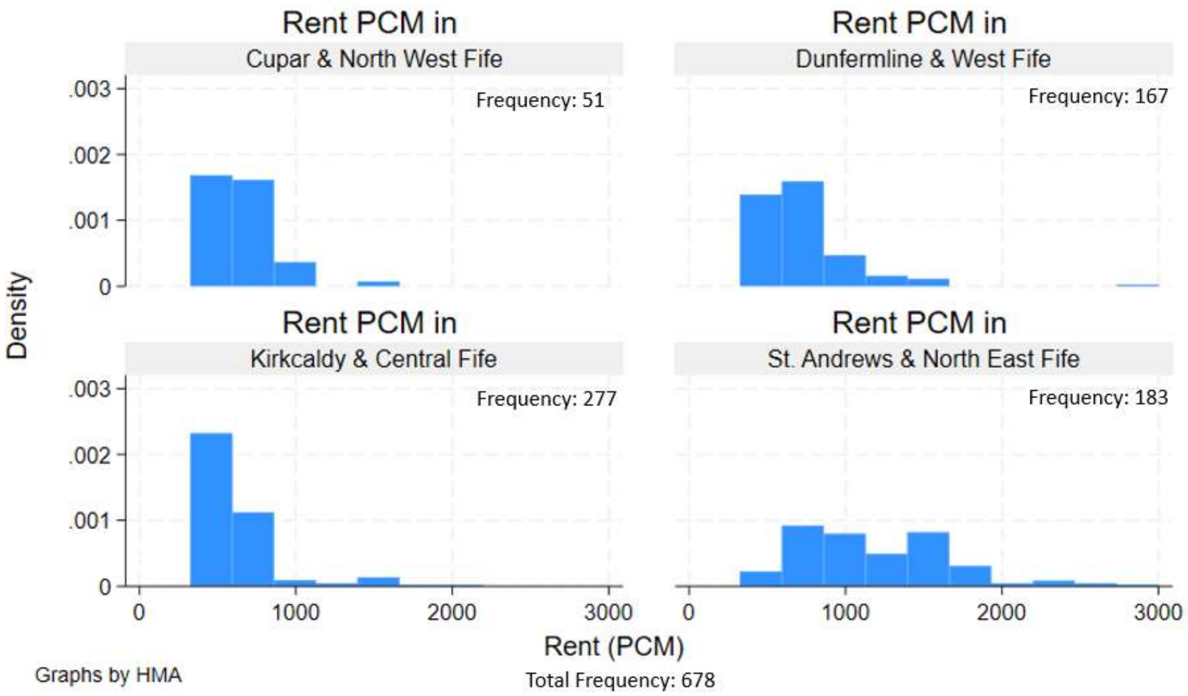


Figure 45: Histograms showing Rent Per Calendar Month in each HMA (Stata code in Appendix 14)

Figure 45 shows histograms comparing the distribution of rent per calendar month, *RentPCM,* in each Housing Market Area. There are four Housing Market Areas in Fife which are used for housing need and demand assessment, and statutory development planning. These were determined through examination where households choose to settle when they move to and within Fife. The majority of properties in this dataset are in Kirkcaldy & Central Fife and St. Andrews & North Fife. Rent in St. Andrews & North East Fife is much more widespread than in the other HMAs. In the other HMAs, the rent is more positively skewed in comparison, and most properties lie within the lower and middle rent ranges.

After looking at the cross-distributions between *RentPCM* and *HMA*, the same was conducted between the variables *RentPCM* and *Town*. In Figure 14, there are 4 histograms. One for Dunfermline, one for Kirkcaldy, one for St. Andrews and the last one for "Other". All towns with less than 15 properties in the dataset were put into the category called "Other" in order to have more clarity when looking at the histograms and to give more statistical significance to the towns with very few observations.

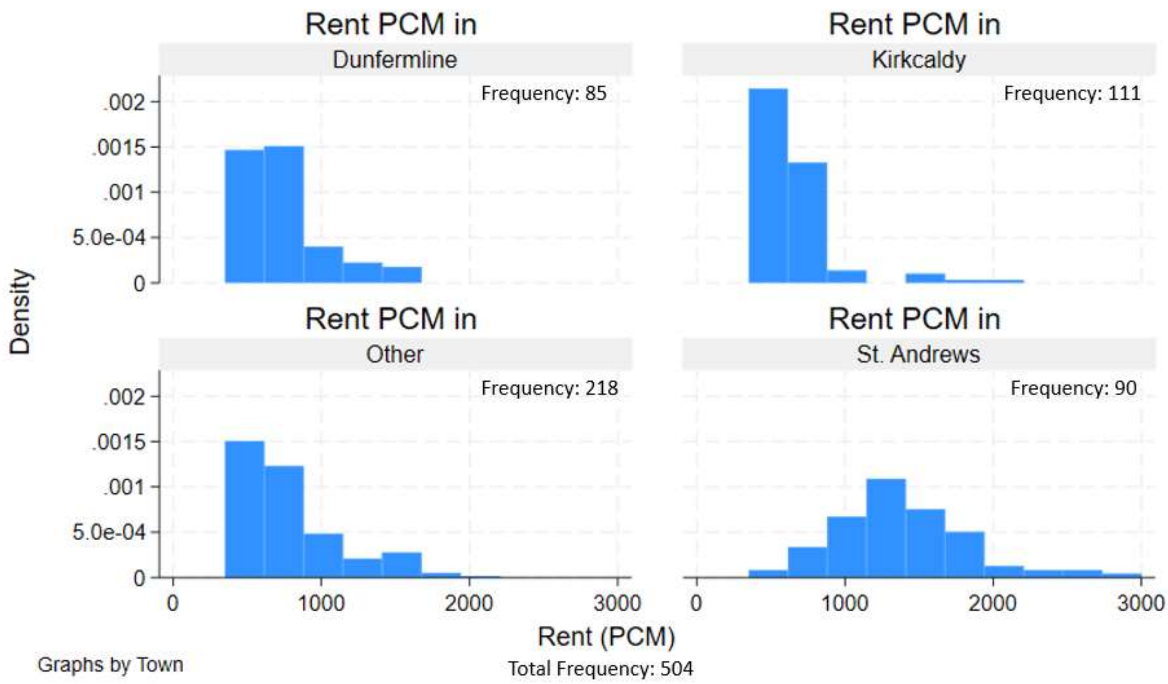


Figure 46: Histograms of Rent Per Calendar Month in different Towns (Stata code in Appendix 14)

The distributions of rent for Dunfermline and Kirkcaldy are very similar to those in Figure 45 for the HMAs Dunfermline & West Fife and Kirkcaldy & Central Fife, respectively. The distribution for the category "Other" is right skewed with most properties having monthly rent below £1000. The distribution for St. Andrews has an almost normal distribution with a longer tail on the right-hand side. St. Andrews has a broader range of rent and most of the properties in this dataset with rent higher than £2000 are in St. Andrews.

We also looked at the rent distribution for each furnishing status which can be found in Figure 47. The distribution of furnished properties shows an even spread across the range of rent. The histogram for part furnished properties has a significant peak at the lower end of the rent range, indicating that most part furnished properties have lower rent. Unfurnished properties show a concentration in lower rent levels with the highest peak among the three categories. From the difference in frequency, part-furnished properties seem to be less common. There is a clear trend that furnished properties tend to have a higher range of rent, likely due to the costs associated with furnishing properties. The varying distributions indicate that the furnishing status of a property is a significant factor in pricing rent.

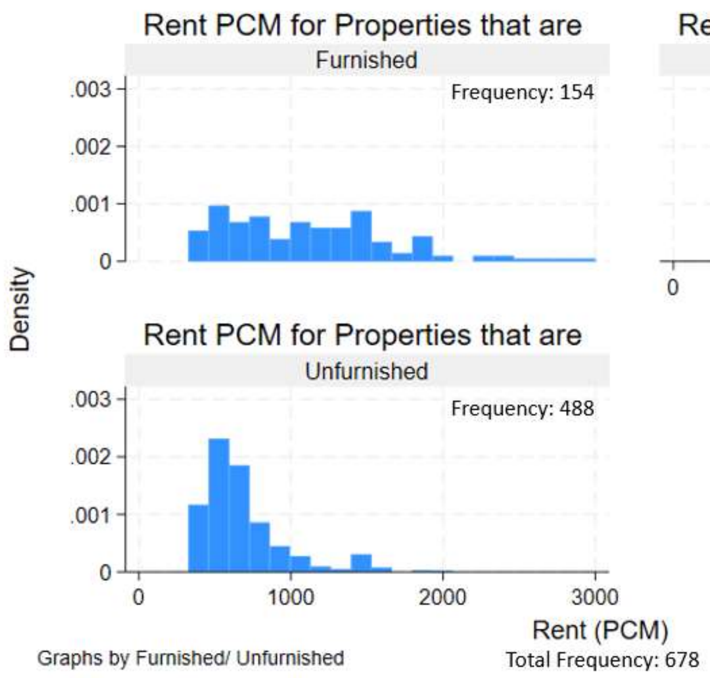


Figure 47: Histograms illustrating Rent Per Calendar Month by Furnishing Status (Stata code in Appendix 14)

Figure 48 depicts two pie charts comparing the furnishing status of properties in Fife and properties in the town of St. Andrews. The majority of properties in Fife are unfurnished (71.87%) while most properties in St. Andrews are furnished (80%). This highlights the difference in rent prices when looking at St. Andrews versus any other town in Fife.

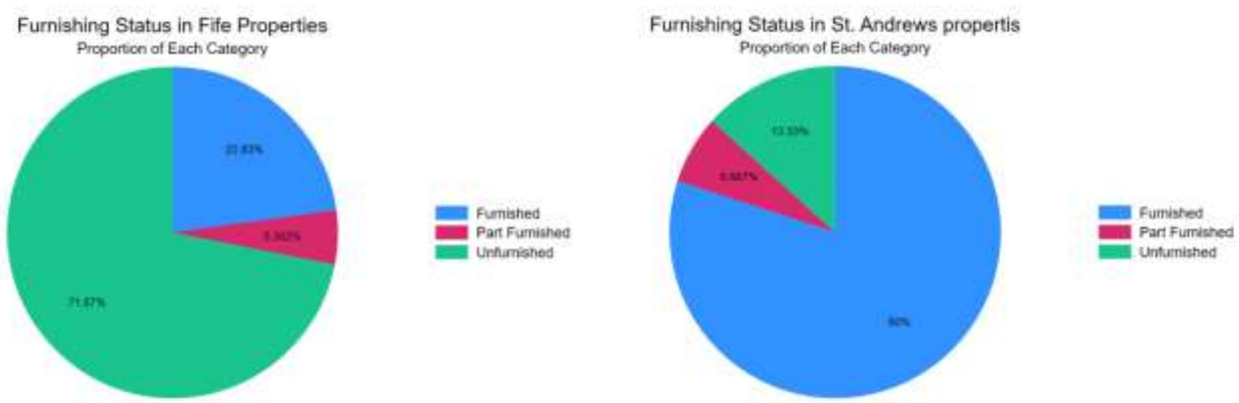


Figure 48: Pie charts comparing Furnishing Status of properties in Fife and properties in St. Andrews alone (Stata code in Appendix 14)

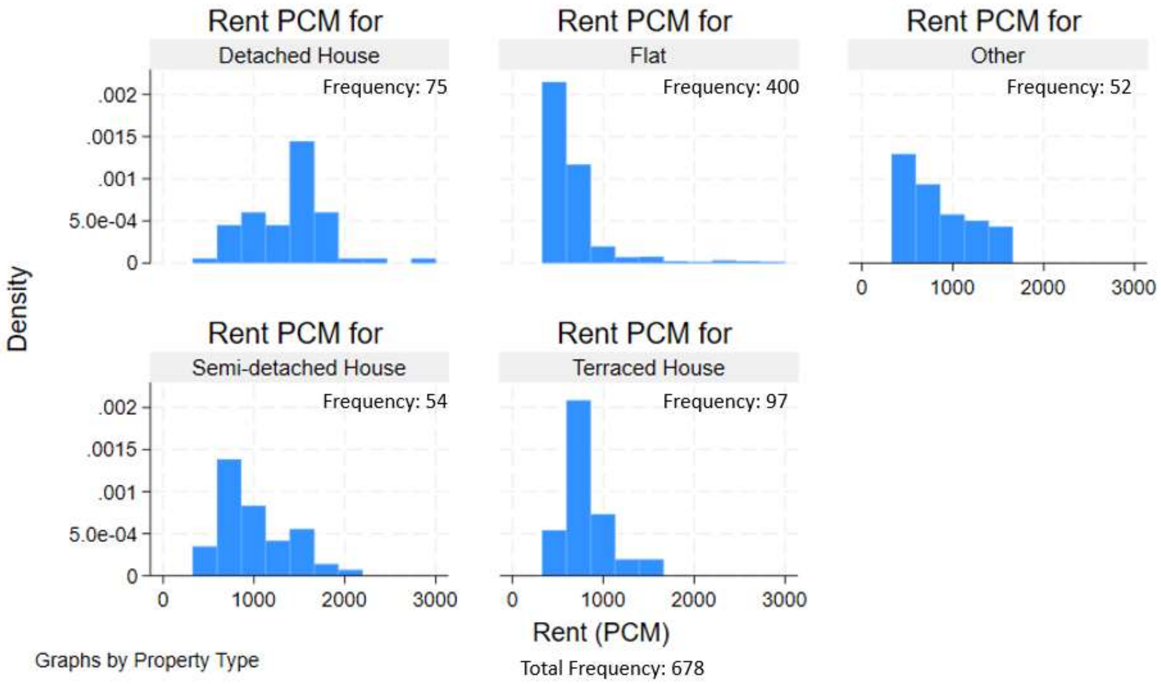


Figure 49: Histograms of Rent Per Calendar by Property Type (Stata code in Appendix 14)

In Figure 49, we looked at the rent distributions for each property type. There are 5 histograms, one for each type: Detached House, Flat, Semi-detached house, Terraced house and Other. In this case, the property types that have less than 20 properties in the dataset were grouped into Other. The most common property types, flats and terraced houses, tend to have lower rents which can be seen by the peaks at lower rent levels. Detached and semi-detached houses have a wider range of rents, which may reflect differences in size, location, and amenities that can drive up rental prices. The Other category has a relatively flat distribution compared to the others, with a slight increase in density at the lower end. This could reflect a mix of less common property types, each with a variety of rental prices.

Lastly the team looked at the distribution of rent over three consecutive academic year: 2021-2022, 2022-2023, and 2023-2024. In Figure 50, there is a visible shift in the distribution of rent prices across the years. Initially, most rents were concentrated at the lower end, but over time, the density spread out, suggesting an increase in both the mid-range and higher-range rent prices.

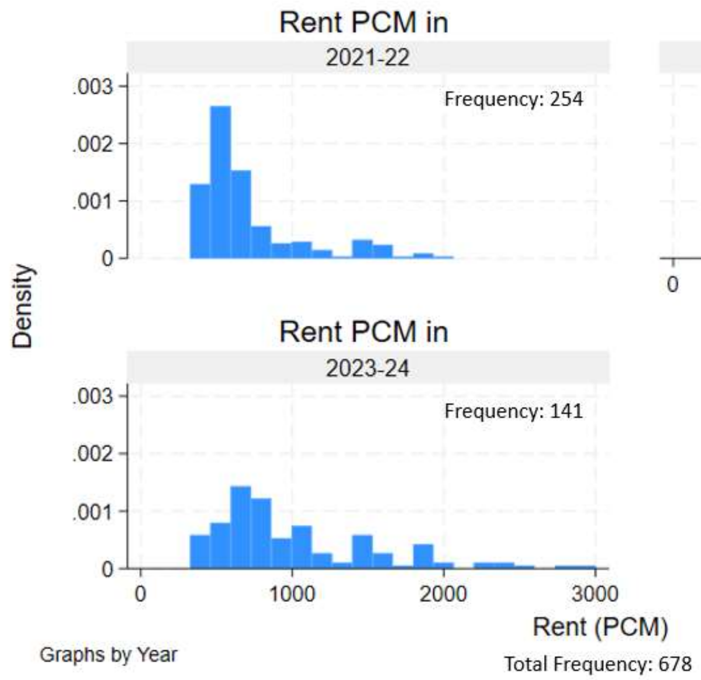


Figure 50: Histograms illustrating Rent Per Calendar Month for each academic year (Stata code in Appendix 14)

4.2 Methods

<u>Descriptive Statistics</u>

The graphical descriptive statistics were produced using Stata. While loops were used extensively in previous semesters to generate descriptive statistics by iterating through all observations and aggregating active licences or bedrooms, this semester saw an improved method utilizing long form data formatting and inbuilt commands for greater robustness. Using the "*collapse (sum) totalActive, by(date)"* command, Stata aggregates the number of active licences or bedrooms for each month from 2011 to 2024, where the variable 'totalActive' contains either the number of active licences or the number of active bedrooms.

<u>Selective Attrition</u>

Drawing on previous semester's hypotheses and methodologies, we continued researching the theory of selective attrition which postulates that property owners and managers differentially sacrifice licences based on property value, location, and occupancy.

We posited that properties with higher occupancy are less likely to sacrifice their HMO licence, given the difficulty of repurposing these properties for non-HMO applications, such as Airbnb accommodation or single-family use. Furthermore, we speculated that the cost of forfeiting a licence could decrease with distance from the town centre, as students might be willing to pay a higher rent premium for centrally located accommodations compared to local families. Consequently, assuming the alternative involves converting the property into a non-HMO rental for a single family, the potential loss in rental income from losing a licence for a property situated in the town centre might be greater than that of a property located on the outskirts of St Andrews. Finally, we hypothesized that property value might inversely correlate with HMO forfeiture, as higher-priced properties generally attract higher rental premiums, making it more challenging to secure long-term single-family renters.

It is important to note that the Overprovision Policy does not actively revoke licences and only restricts the issuance of licences to previously unlicenced properties, stating that pre-existing licences can continue to be licensed if they maintain the relevant safety standards. Therefore, changes in licence supply solely reflect the choice of property owners and managers.

From the register dataset, we created a new variable called 'lost_licence' that served as our dependent variable. 'lost_licence' is a binary variable assigned to "1" if a property held an active licence in 2018 but not in 2024. Additionally, we created three binary occupancy variables called 'four_bed', 'five_bed', and 'six_beds_plus' that equal one when the property has a licenced occupancy equal to 4, 5, or greater than 6, respectively. While the minimum number of occupants for an HMO licence is 3, a 'three_bed' variable was not created as it would induce perfect collinearity among the occupancy variables. In this way, the constant value in the regression relates to a 3-bed property.

The last variable included in the regression to explore the location effect was the 'DistancetoTown' variable which provides the walking distance from each postcode to 100 Market Street, which we consider to be the amenity centre of St Andrews.

We tested these hypotheses using several different functional form variations of a linear probability model (LPM). The general form of the LPM is given below.[7]

$$P(y = 1\,|x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

The model is used to estimate the probability of an event occurring as a linear function of the explanatory variables. For example, β_1 is interpreted as the change in the probability of the event occurring given a one-unit increase in x_1.

The specific specification of model used for our analysis is given below.

$$lost\_licence\_2018\_2024 = \beta_0 + \beta_1 \boldsymbol{four\_bed} + \beta_2 \boldsymbol{five\_bed} + \beta_3 \boldsymbol{five\_beds\_plus} + \beta_4 \boldsymbol{AdjPrice\_HPI} + \beta_5 \boldsymbol{DistancetoTown}$$

[7] Wooldridge, J. (2010), "Econometric Analysis of Cross Section and Panel Data", Second Edition: The MIT Press

After finding no statistically significant results using the basic regression, we opted to explore another functional form that includes squared terms for the property value (AdjPrice_HPI) and distance (DistancetoTown) variables. The addition of squared regressors for the distance and property value variables allows the model to capture non-linear relationships between the predictors and the response variable, thereby improving its flexibility and possibly producing statistically significant results. The revised regression equation is given below.

$$lost\_licence\_2018\_2024 = \beta_0 + \beta_1 \boldsymbol{four\_bed} + \beta_2 \boldsymbol{five\_bed} + \beta_3 \boldsymbol{five\_beds\_plus} + \beta_4 \boldsymbol{AdjPrice\_HPI} + \beta_5 \boldsymbol{AdjPrice\_HPI^2} + \beta_6 \boldsymbol{DistancetoTown} + \beta_7 \boldsymbol{DistancetoTown^2}$$

4.3 Results

<u>Selective Attrition Results:</u>

The output of the basic selective attrition regression is provided in figure 51 below.

```
. reg lost_licence_2018_2024 four_bed five_bed six_beds_plus AdjPrice_HPI DistancetoTown if active20
> 18 == 1

      Source |       SS           df       MS      Number of obs   =       310
-------------+----------------------------------   F(5, 304)       =      1.73
       Model |  1.98209395         5  .39641879   Prob > F        =    0.1284
    Residual |  69.8275835       304  .229695998   R-squared       =    0.0276
-------------+----------------------------------   Adj R-squared   =    0.0116
       Total |  71.8096774       309  .232393778   Root MSE        =    .47927

------------------------------------------------------------------------------
lost_li~8_2024 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
---------------+----------------------------------------------------------------
      four_bed |  -.0500789   .0629605    -0.80   0.427    -.1739724    .0738147
      five_bed |   .1316785   .0785579     1.68   0.095    -.0229075    .2862645
 six_beds_plus |  -.0929378   .1692215    -0.55   0.583    -.4259315     .240056
  AdjPrice_HPI |  -2.23e-07   1.67e-07    -1.33   0.183    -5.52e-07    1.06e-07
DistancetoTown |   .0288323   .0648544     0.44   0.657    -.0987881    .1564528
         _cons |   .4328639   .1136221     3.81   0.000     .2092786    .6564492
--------------------------------------------------------------------------------

.
```

Figure 51: Base Selective Attrition Regression Output

The regression has an extremely low R^2 and Adj-R^2 of 0.0276 and 0.0116, respectively, indicating that the model explains essentially none of the variance in the 'lost_licence' variable. Additionally, none of the regression coefficients were statistically significant at the 5% confidence level and the F-statistic for overall significance of 0.1274 is not statistically significant at any commonly accepted significance level. However, coefficient for the 'five_bed' variable is marginally statistically significant at the 10% level (p-value = 0.095). The coefficient of 0.1317 would suggest that a five-occupant property is approximately 13% more likely to be sacrificed when compared to a three-occupant property.

As outlined in the methods section, after recognising statistical insignificance in the base model, we explored a different functional form with added squared terms for the distance and property value factors. The output of that regression is provided in Figure 52 below.

```
. reg lost_licence_2018_2024 four_bed five_bed six_beds_plus AdjPrice_HPI AdjPrice_sqr
> DistancetoTown DistancetoTown_sqr if active2018 == 1

      Source |       SS           df       MS      Number of obs   =       310
-------------+----------------------------------   F(7, 302)       =      1.59
       Model |  2.54489439         7  .363556342   Prob > F        =    0.1392
    Residual |   69.264783       302  .229353586   R-squared       =    0.0354
-------------+----------------------------------   Adj R-squared   =    0.0131
       Total |  71.8096774       309  .232393778   Root MSE        =    .47891

------------------------------------------------------------------------------------
lost_licenc~8_2024 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------------+----------------------------------------------------------------
          four_bed |  -.0394471   .0634292    -0.62   0.534    -.1642663     .085372
          five_bed |   .1394601   .0789776     1.77   0.078     -.015956    .2948761
     six_beds_plus |  -.1416329   .1719628    -0.82   0.411    -.4800298    .1967641
      AdjPrice_HPI |  -8.57e-07   4.90e-07    -1.75   0.081    -1.82e-06    1.07e-07
      AdjPrice_sqr |   5.49e-13   4.10e-13     1.34   0.181    -2.58e-13    1.36e-12
    DistancetoTown |  -.1332391   .1878518    -0.71   0.479    -.5029032    .2364251
DistancetoTown_sqr |    .072092   .0844405     0.85   0.394    -.0940742    .2382582
             _cons |   .6493983   .1791325     3.63   0.000     .2968924    1.001904
------------------------------------------------------------------------------------
```

Figure 52: Selective Attrition with Squared Terms Output

Similarly to the base model, the regression above is strikingly statistically insignificant and has low explanatory power. The low R^2 and Adj-R^2 of 0.0354 and 0.0131, respectively show that the model still does not explain almost any variance in lost licences. Again, none of the coefficients are statistically significant at the 5% level, and the F-statistic for overall significance is even higher at 0.1392.

Fife Council Statistical Discrepancy

In May 2023, Fife Council shared quantitative research on the impact of the Overprovision Policy during a meeting of the Cabinet Committee in which they voted on whether to renew the policy for a further three years. The Committee voted to renew the policy with an agreed exemption to grant up to 15 new HMO licenses for properties managed by the University St Andrews, based on the understanding that the total supply of HMO properties had decreased by 17 since the introduction of the policy, corresponding to a 124 person decline in occupancy. The specific statistics outlined in the meeting agenda are presented in Figure 53.

Table 1: HMO Licences (Granted & Pending) 2019 and 2023 (St Andrews Town)

HMOs in St Andrews 07/03/2019		HMOs in St Andrews 14/02/2023	
Total No. HMOs	Total Permitted Occupancy	Total No. HMOs	Total Permitted Occupancy
1,046	6,994	1,029	6,870*

*4,272 is the occupancy of student halls, Table 2 shows the full breakdown

Figure 53: *HMO supply statistics cited by the Fife Council Housing Services department during the Overprovision Policy Review Cabinet Committee meeting in May 2023*

However, the supply statistics that we obtained from the HMO register provided by Fife Council differ significantly from those outlined above.

	Total Active HMOs		Total Occupancy	
	March 7, 2019	February 14, 2023	March 7, 2019	February 14, 2023
Council Figures (Granted & Pending)	**1,046**	**1,029**	**6,994**	**6,870**
Change		*-17*		*-124*
% Change Over Period		*-1.6%*		*-1.8%*
Our Findings (Granted)	**947**	**740**	**6,225**	**4,671**
Change		*-207*		*-1,554*
% Change Over Period		*-21.9%*		*-25.0%*

Figure 54: Table Comparing our Statistics with the Council

As shown in Figure 54, the Council identified 17 (-1.6%) lost licences and 124 (-1.8%) lost beds between March 7, 2019, and February 14, 2023. Contrastingly, our findings suggested that 207 (-21.9%) properties lost their license during the same period, corresponding to 1,554 (-25%) lost beds.

The primary reason for the divergence is our differing definitions of HMO supply. We defined supply as the number of active HMO licenses at any given time, according to the HMO register. On the other hand, Fife Council considered supply equal to the number of active licenses plus the number of properties with an expired license and pending renewal application. To seek clarity on the matter, we spoke to the HMO licensing department at Fife Council who confirmed that the conditions of a previous license remain in force until the renewal application is processed and a determination is made on the application. This provision is outlined in Section 135 of the Housing (Scotland) Act 2006.

As the register provided to us contained pending applications on the date it was exported (9 February 2024), we were able to replicate the Council's methodology and suggest whether their calculations were reasonable. We calculated 1,079 pending and granted licenses active on 9 February 2024 (868 active and 211 pending), corresponding to 7,092 occupants. We identified 70 duplicates in cases where a property with an active license had also applied for a renewal license, reducing the number of pending and granted licenses to 1,044, corresponding to 6,942 occupants. These supply figures are between the numbers cited by Fife Council for March 2019 and February 2023. Therefore, we have evidence that the methodology used by Fife Council to tabulate HMO supply is reasonable and robust.

4.4 Discussion

Selective Attrition

We were unable to prove the hypothesis that property owners and managers differentially sacrificed licences based on property value, location, or occupancy between 2018 and 2024. As discussed in the results section, the Adj-R^2 values for the base regression and squared terms regressions of 0.0116 and 0.0131, respectively, indicate that both functional forms explain almost none of the variance in the data during the period. Furthermore, all regressions produced statistically insignificant regressor coefficients at the 5% significance level suggesting that none of the studied property characteristics meaningfully explain whether properties lose their licence.

Our findings mirror those from the Martinmas Semester 2023/24 Progress Report 2[8], which did not identify any statistical significance when running similar regressions on the faulty 'Kategorical' dataset.

After replicating the selective attrition regression on the reliable data provided by the Council which yielded similar negative results, we conclude that selective attrition, as theoretically developed here, is not a significant characteristic of the St Andrews HMO rental market.

As discussed above, the HMO register does not contain information on the number of pending licenses across time. Therefore, our active supply statistics do not include properties with an expired license but pending renewal application. Consequently, the 'lost_licence' variable likely overstates the number of lost licences as many of the licences it considered inactive in 2018 or 2024 were likely pending. As such, our regressions employ faulty data which largely invalidates the results. In the case that future teams gain access to accurate granted and pending data, it may be worthwhile to replicate the selective attrition model using the accurate supply data.

Future Work

We hope to obtain monthly pending HMO application data from Fife Council through Freedom of Information over the summer, future teams will likely wish to recalculate the monthly HMO supply statistics by adding pending applications and active licenses. This information will be critical for the work of other teams.

Separately, we believe the next logical step for register-related research involves analysing the short-term let (STL) register due to the scale of short-term letting that we have identified this semester. As discussed in the relevant section, future teams may wish to improve the HMO register / STL register address matching process. Subsequently, participants may attempt to identify the rental status of properties that match between the registers (possibly through door-to-door surveying) to identify if the HMO supply statistics are inflated. Several other interesting questions could be investigated, including:

[8] Carlsen et al. (2023), "Housing in St. Andrews: 2023/24 Martinmas Semester Progress Report 2"

- Is it more profitable for landlords to rent properties as HMOs or short-term lets?
- What factors influence the decision of landlords to rent properties as short-term lets or HMOs?
- How has the supply of short-term let properties evolved over time?
- How seasonal is the demand for short-term lets?
- What is the average occupancy rate of the short-term let housing stock?

5. Supply Team

The Supply Team aimed to understand the determinants of rental property prices in the town of St. Andrews, Fife, through an examination of hedonic pricing models and regression-based analyses. The main objective was developing a price index that accurately represents yearly increases in inflation-adjusted rental prices over time.

A price index tracks changes in the overall level of prices for goods and services over time, enabling individuals to understand how their real income has changed, accounting for inflation. The same concept extends to rental property prices, as a property price index can aid individuals in understanding how rent prices have changed over time, providing insights which may be useful for local policymakers or stakeholders. To assess changes in rent prices in St Andrews over time, we collected data on rental prices through historic listings on property websites using an internet archive, "Wayback Machine", to construct a longitudinal dataset for the years 2012 – 2024. Using this database, we conducted hedonic pricing regressions to examine the key determinants of St Andrews' rental property prices, as well as a price index function to examine real changes in prices over time. Our key findings were i) a £3.69 increase in monthly rental price per room each year, ii) a U-shaped relationship between number of rooms in a property and rent price per room with rooms in 3-bedroom properties commanding the lowest price, and iii) a decline of £24.05 in monthly rental price per room for each kilometre increase in distance from the centre of town, as defined by Tesco on Market Street.

The Supply Team members are Tasha Delvecchio, Bahrathi Keeping, Phoevos Kreps, and Tara Nair.

5.1 Data

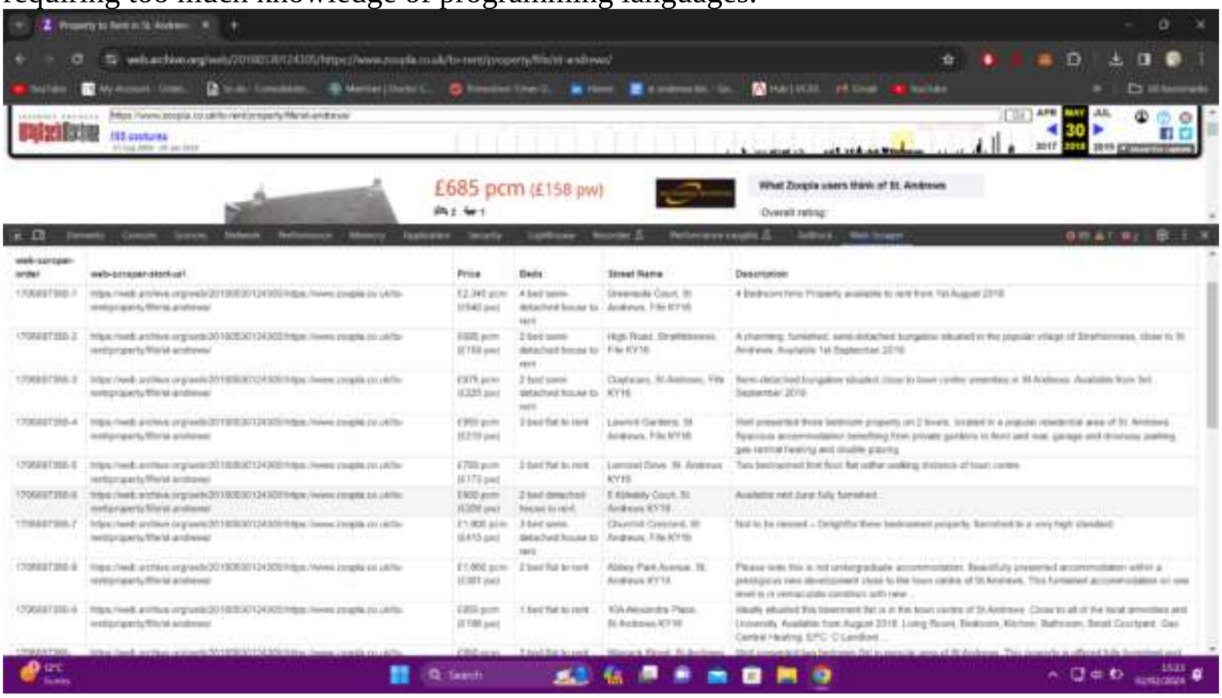
<u>Web Scraping</u>

To construct a longitudinal dataset on St Andrews rental property prices, we made use of an internet archive called Wayback Machine, which record snapshots from websites over time. To record all the snapshots efficiently, we used a technique called web scraping, which refers to the process of extracting content from websites. We made use of Python to extract longitudinal data from the years 2012-2018. The full code used to scrape longitudinal data is shown in **Appendix 15**. When scraping data, we are extracting data from HTML (Hyper Text Markup Language), so we needed to use various packages and libraries available in Python to fetch all the components of the data needed. Most of the code was the same as the web scraping code used by the Housing St Andrews team in the 2021-2022 Candlemas Semester. We scraped Zoopla listings because nearly all real estate agents listed properties on Zoopla, so it served as an aggregate website for us to scrape data from. Furthermore, alternatives such as Rightmove had far fewer properties listed. Zoopla's website layout changed in 2018, so the code would return an error if we tried to run it for the years after 2017, thus we added a try-exception block to ensure the code still generated the database for properties until 2018. We used the code in **Appendix X2** to scrape HousesForSaleToRent from the Internet Archive for the 2020 to 2022 period. This Python script acts analogously to the script in **Appendix X;** however, it is adapted for the different layout of HousesForSaleToRent rather than Zoopla. We used the code to obtain the address of the property, the rent, the number of bedrooms, and the exact date of listing.

We also generated a cross sectional price index, which involves calculating how prices vary according to their location in St Andrews. We scraped websites such as Zoopla (www.zoopla.co.uk), HousesForSaleToRent (www.housesforsaletorent.co.uk), Thistle Property as well as Facebook pages of real estate agencies. We used the Tesco store at 138-140 Market Street as a central point, as it's located on the main commercial street in the central part of the town. We then manually calculated the walking distance from the property to Tesco. Since some webpges simply listed the street address and not the number of the property nor the latter half of the postcode, we calculated the distance from the centre of the street to Tesco using the coordinates given on Google Maps.

<u>Alternate methods</u>

Although coding was our dominant web scraping method, we used other approaches too. This was done for two reasons- firstly, not all team members were as experienced in coding, and secondly, certain websites and sources were challenging to scrape data from using traditional coding methods. The first alternate method used a developer tool web scraper software, which is accessible via a chrome extension known as 'Free Web Scraper'. To work the tool, we programmed element and text selectors to instruct the scraper on how to navigate the website (such as The Wayback Machine) and which data to extract. We programmed the software to obtain address, pricing, bedroom, and description data, which the software expressed into a table (as shown below in Figure A1), and then into Excel, which was repeated for each year/snapshot. Overall, this method allowed for efficient, automated web scraping, whilst not requiring too much knowledge of programming languages.



<u>Figure 55. The Developer tool Scraper Software extracting data from the Wayback Machine</u>

Another method was manually scraping data from Facebook and other sources, which we used for years where the Wayback Machine was lacking in data (e.g. 2019). In these instances, due

to the variation in the wording/structure of posts, we could not reliably extract data using an automated process such as coding or the developer tool scraper software. This meant manual scraping was the most effective option, and although tedious at times, was rather straightforward as it simply involved copying and pasting information from posts into the database.
The final method we used to gather data was contacting rental agencies within St Andrews and requesting any sort of pricing information they had. Unfortunately, this method did not prove useful, and despite reaching out to around 14 agencies, we only received 2 responses, and neither ended up being willing to provide us with any data. We believe this is likely attributable to confidentiality issues surrounding sharing such data, as well as agencies being occupied with their own business operations, and therefore not having the time to assist us.

Improvements to the dataset

Once we gathered all the data, we edited the dataset to improve its reliability and validity. The first improvement we made was to increase the sample for certain years where we felt the number of properties was inadequate, and therefore could not accurately represent the St Andrews rental prices of that given year. To address this, we scraped more properties for these years using untapped sources of information. This included the likes of the Alba residential Facebook Page, as well as revisiting the Wayback Machine but using other archives of websites such as Rightmove instead of Zoopla. Using a mix of the developer tool scraper software and manual methods, we were able to scrape over 50 new properties, which resulted in a more comprehensive and representative dataset, giving us greater confidence in our results.
Another improvement made to the dataset was removing duplicate properties which had arisen due to the extraction of listings from multiple sources. In defining duplicates, it was important to identify true duplicates while minimising falsely flagged properties – those with similar details but which were in fact different properties. As such, we defined duplicates as properties with the same price, address, number of bedrooms and listed within 6 months of each other. To remove most of the duplicates, we used the *CONCAT* function in Excel to create a new column with a unique identifier based on the property row data- or specifically, the price, bedroom and address information. Having created the new row, we used conditional formatting to flag duplicate identifiers which signalled that the price and address information was the same for two or more properties. We then manually reviewed the information and removed the duplicate if it had been listed within 6 months of the other property. Finally, we manually reviewed the entire dataset to ensure no properties had been missed due to entry errors such as differences in address spelling. This ultimately resulted in the removal of over 20 duplicate properties, ensuring our dataset was as clean and accurate as possible.

Dataset Analysis

We have a total of 929 properties in our dataset, including every snapshot from the Wayback Machine that was listed. The properties listed vary in the month of listing throughout each year, but we have properties from nearly every month for each year. The figure below shows the number of properties sampled per year.

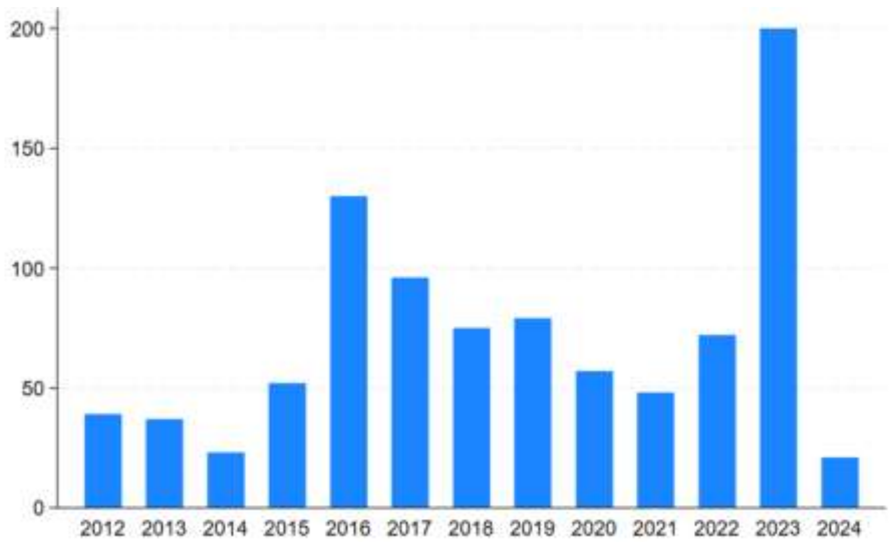


Figure 56. Bar chart showcasing the sample of properties per year 2012-2024

Even though the Wayback machine acted as a quick and efficient method of scraping data, we acknowledge its limitations. One issue with using online sources like the Zoopla archives is the sparse data available in earlier years. For example, as evident in Figure 56, though more recent years typically featured around 50+ properties (with the exception of 2021 at the height of the pandemic), we managed to scrape only 23 properties in 2014. This limitation in data availability means our sample of properties for earlier years may be limited in how representative they are of the true rental market. Another issue with the sampling method is that in each snapshot, most properties would have been on the market for some time, at least before the moment of observation. For example, in 2012, most of the properties were samples from September, when there is a large inward migration due to the start of the academic year. As such, it is possible the properties still present on the market in September were not representative of all private rental properties in the period, e.g. they may the most expensive ones, which remained on the market after the cheaper ones were rented out. More generally, properties which remain on Zoopla for longer periods of time are more likely to appear in snapshots. This could mean our sample is biased towards less desirable properties, e.g. those which are the most expensive, which could limit the representativeness of our database.

Furthermore, we also acknowledge the potential for error having scraped from rental agencies' listings, as we recorded the price as the actual paid monthly rent, which may not always be the case. We have not sampled the reported rent price by the tenants, and it is possible that after negotiations the actual price paid by the tenants may have differed from the price advertised online. An additional issue with the specific use of the Wayback machine was an error in the Zoopla 2019 snapshot. The Wayback machine states that the website was saved and recorded in 2019, but the links for 2019 turn out not to show the website in the claimed date, but instead redirect to a snapshot from either 2018 or 2020. This meant we had to manually double check each snapshot date to ensure the Wayback engine saved a copy of the Zoopla website on the correct date, hence we had to collect 2019 values solely from other sources.

Leading on from this, we recognise that the range of sources we used in our data collection (Wayback Machine Zoopla, HousesForSaleToRent, Facebook posts etc.) may have resulted in inconsistencies or biases throughout the dataset. For example, the data from Facebook posts could arguably be considered slightly more 'informal' than data from more reputable sources like Zoopla, and therefore may have more inaccuracy in price. Given that we did not take a proportional number of properties from each source per year, it is possible that this may have had an impact on the reliability of our findings, which we believe is important to consider when drawing meaningful conclusions.

Descriptive Statistics for Price Index

Once the team were content with the size of the data sample (over 900 observations), the dataset was imported into Stata to produce descriptive statistics. The descriptive statistics were used to identify anomalous observations, to ensure reliability of our dataset. Without being correctly identified, outliers in the dataset could skew our results or misrepresent the underlying rental market dynamics. The first step was to generate a time variable that transforms the date from which the property was put online into the respective *Year* variable in order to look at yearly trends. After doing so, the team investigated the monthly rent distribution over our time period of 2012-2024.

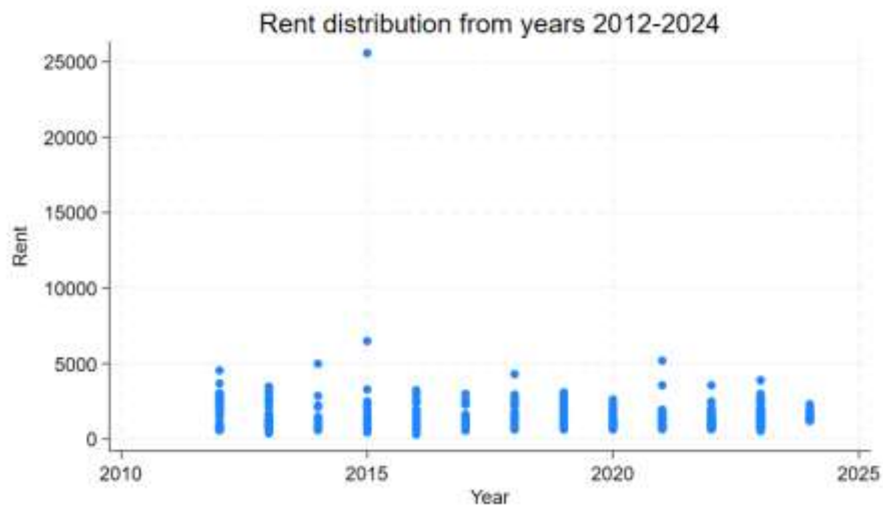


Figure 57. Scatter plot of initial rent data over years 2012-2024

The distribution of rent in Figure 57 shows an obvious outlier in 2015 disturbing the range of our monthly rent distribution, almost 5-times larger than the second largest rent price observation. Instead of dropping the large outlier, the team decided to proceed differently and trim the top 5% and bottom 5% of the monthly rent observations to mitigate the effects of outliers on our analysis and approximate a more normal distribution. In Figure 58, the rent distribution has been trimmed and offers a more appropriate range for our monthly rent data; the average rent lies between £500 and £3000 per month, but some rent values can reach up to £5000 per month.

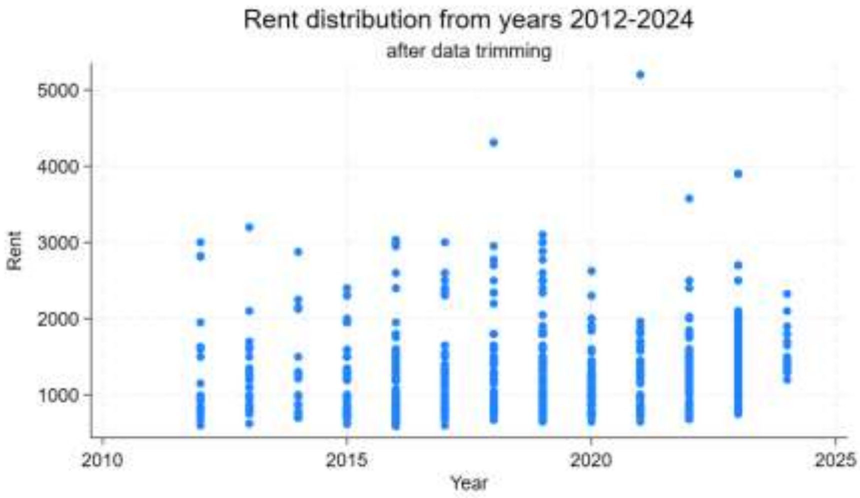


Figure 58. Scatter plot of monthly rent distribution after data trimming

As our main goal is to form a price index from hedonic pricing results, the aim is to be able to quantify the rate of change of monthly rent over our sample time period. To get an accurate result, it is extremely important to account for inflation that occurs over the years. Without adjusting for inflation, possible issues can arise: the overestimation of growth, misleading the financial aspect of the analysis, and inaccurate comparisons across time. To adjust for inflation, the year 2012 was selected as base year as it is the first year in our data. Using the consumer price index (CPI) published by the Office of National Statistics, deflator coefficients were generated for each year, as seen in the table below.

Years	Deflator coefficients
2012	1
2013	1.02643
2014	1.04651
2015	1.04968
2016	1.0518
2017	1.07188
2018	1.10359
2019	1.12368
2020	1.14376
2021	1.15222
2022	1.21459
2023	1.33615
2024	1.39006

Table 1: Deflator coefficients used to adjust rent prices to inflation (base year:2012)

The deflator coefficient was constructed by taking the decimal value of the percentage change in the CPI[9] from 2012 to 2013 and adding 1 to it. This value would then be the deflator coefficient for the year 2013 and will divide all rent values of properties from 2013 to produce the inflation-adjusted monthly rents. The same process was repeated for every year. The prices from 2012 remain unchanged since it is the base year, hence the deflator coefficient is 1. A new variable *AdjustedRent* stores the inflation-adjusted rents per month.

One of the main variables in our dataset is the number of bedrooms per property. Since it is likely that properties with more bedrooms have higher rent than properties with fewer bedrooms, the team was interested in exploring the monthly rent per person values. To do so, a new variable *AdjustedRentPP* takes the inflation-adjusted rent and divides it by the number of bedrooms in the property. This was done under three assumptions. Firstly, we assume that there is one person per bedroom, hence this report uses the terms “rent per person” and “rent per room” interchangeably. Our final two assumptions are that the property’s total rent price is split equally between occupants, and that all bedrooms are identical apart from along the dimensions identified by our other variables like distance to town, total rooms in a property, etc.

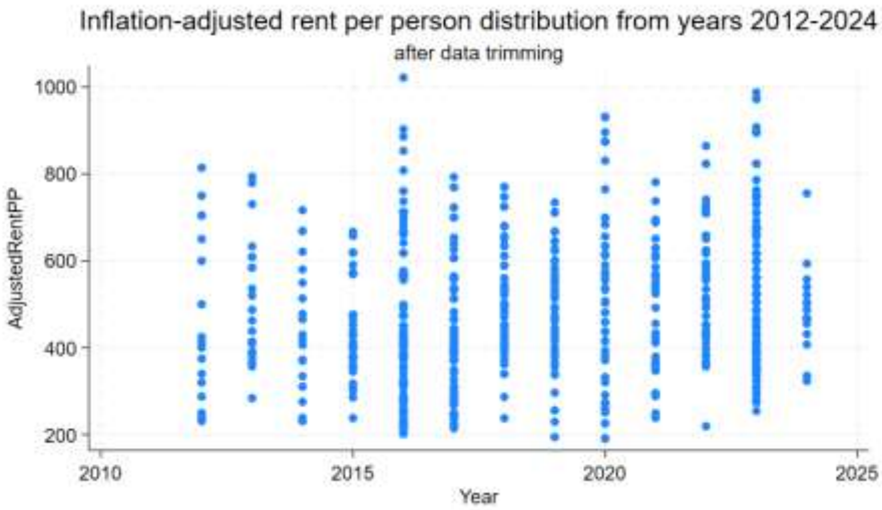


Figure 59 Scatter plot of inflation-adjusted rent per person distribution from 2012 to 2024

Figure 59 represents the inflation-adjusted rent per person distribution. This was the last step for the descriptive statistics as now the data has been prepared and is ready to be used to generate hedonic price regressions. The monthly rent per person ranges from £200 to slightly over £1000 as shown in Figure 59.

[9] “CPI Inflation Calculator UK: CPI Calculator Data from 1988.” *CHARTERED SURVEYORS LONDON*, 22 June 2021, erikasgrig.com/calculators/inflation-calculator-cpi/.

Fife Council Rental Price Data

In addition to the data from web scraping, we conducted some minor analyses on a dataset provided by Fife Council, collected by the Register Team. This dataset includes properties in Fife rented over three academic years (2021-2022 to 2023-2024) with information on the property type, rent per calendar month, location, number of bedrooms, year the property was rented, and furnish status. However, given that this data covers such a short-term period and most importantly that Fife Council did not provide clear guidance as to their data collection methods, we have relied minimally on this data to produce results.

5.2 Methods

Descriptive Statistics

Given that the initial information we had in our data was rent per calendar month, number of bedrooms, the address, and the date the property was uploaded onto Zoopla with use of the Wayback Machine, the data needed to be worked on as mentioned in the section above. The descriptive statistics and graphical illustrations were computed using the Stata software.

Hedonic Pricing Regressions

From our newly adjusted dataset, we started constructing hedonic pricing regressions that would estimate the effects of various factors in the rental market in St Andrews and provide quantitative results. The aim is to statistically predict rent, serving multiple purposes such as descriptive analysis to understand property valuation, providing an input for other analyses, and facilitating discussions on mispricing and student preferences regarding property features. The hedonic price regression uses the monthly rent per person adjusted for inflation, *AdjustedRentPP*, as the dependent variable. A linear time trend variable was generated, *yearsince2012*, to observe the change in rent over the years starting at 0 for the year 2012. The variable, *Rooms*, is a discrete variable representing the number of bedrooms in the property. In the regression *Rooms* is not used as a independent variable, but instead a series of dummy variables, *RoomDummy**, each one representing a different number of bedrooms. The regression includes the option *noconstant*, to remove the constant coefficient term. Including this option allowed us to incorporate all the room dummy variables into the regression without raising any multicollinearity issues. It also allows us to interpret the coefficients of the room dummy terms as the rent price per room for a room in a property with a given number of bedrooms, rather than as a premium relative to a base category. This is particularly appropriate as room dummy 1 would have been the natural base category, but rooms in 3-bed properties are in fact the cheapest (i.e. all other rooms essentially command a premium over a room in a 3-bed). The *CentredDistance_weighted* variable was generated from another variable, *DistancetoTown,* which measured the distance of the property from the centre of town. The local supermarket Tesco Express (138-140 Market Street) on the main street of St Andrews was chosen as the centre of town. From this, we calculated the weighted mean distance with respect to rooms. The *CentredDistance_weighted* variable takes the *DistancetoTown* value minus the weighted mean distance. The purpose of this centring process is to ensure that the variation in rent prices attributed to *DistancetoTown* is not confounded by the overall average distance. It allows for a more precise examination of how changes in distance influence rent prices, independent of other factors. Our regression model is as follows:

$$\boldsymbol{Adjusted\ Rent\ per\ person} = \beta_1 \boldsymbol{yearsince2012} + \beta_2 \boldsymbol{RoomDummy1} + \beta_3 \boldsymbol{RoomDummy2} + \beta_4 \boldsymbol{RoomDummy3} + \beta_5 \boldsymbol{RoomDummy4} + \beta_6 \boldsymbol{RoomDummy5} + \beta_7 \boldsymbol{RoomDummy6} + \beta_8 \boldsymbol{CentredDistance\_weighted}$$

The equation above represents the regression used as our main result. However, the team generated other regressions, including different interaction terms, a second time trend variable, and different location for a measure of distance. A second regression was generated replacing the dependent variable, *AdjustedRentPP*, with its natural logarithm form. The decision to use the logarithmic form allows us to linearize non-linear relationships and simplify the interpretation of coefficients as elasticities, it helps manage skewed data distributions and reduce heteroscedasticity, enhancing the robustness and reliability of the regression results. Another regression specification included a second time trend variable, constructed in the same manner as the variable *yearsince2012*. This new variable, *yearsince2020,* was generated to observe the change in rent over the years starting at 0 for the year 2020. The choice of starting the trend from the year 2020 was to evaluate any new possible trends in the rental market that could have been caused due to the Coronavirus pandemic. A different regression included an interaction term *Rooms* x *DistancetoTown* to observe if these two variables jointly influence monthly rent. A new distance variable was also generated, in the same manner as the variable *CentredDistance_weighted*. This new variable uses the location of the University Sports Centre that is located further from the centre of St Andrews. This variable would allow us to interpret how rent varies in different parts of St Andrews. Sample code for the generation of the weighted distance to the University Sports centre can be found in Appendix Item Y2.

Price Index

Using the output of our regression, we then were able to construct a price index function. In generating a price index, the focus shifts towards understanding rent variations over time, while filtering out differences resulting from sampling various properties in different years. Here, the emphasis lies on producing a clean and accurate price index, with particular attention to the time trend coefficient.

Using Stata, we computed a price index adjusted for the base year 2012. Firstly, we initialised two scalar variables, *sum_weights*, and *p_2012_adj*, both set to zero. These variables are crucial for calculating the weighted sum of coefficients and the adjusted price index for the base year 2012, respectively. Subsequently, the code iterates through each room dummy variable, calculating the proportion of properties falling into each room category and computing the corresponding weight by multiplying the proportion with the category number (e.g. if we are referring to *RoomDummy2* the proportion would be multiplied by 2). These weights are then used to update the sum of weights variable and to determine the adjusted price index for the base year 2012. After normalizing the adjusted price index by dividing it by the sum of weights, a new variable, *p_year_adj*, is generated to calculate the adjusted price index for years other than 2012. This calculation involves adding the product of the time trend coefficient, *_b[yearsince2012]*, and the number of years since 2012 to the adjusted price index for the base year 2012. To visualise the price index, a line plot is used. The code for this price index can be found in Appendix Y1.

Fife Council Rent Dataset Hedonic Pricing Regressions

Hedonic pricing regressions were likewise carried out using Fife Council's rent dataset. This required generating dummy variables for based period years, property type and property location. We also used a different outlier trimming method, which was to exclude observations whose adjusted monthly rental price was greater than double the mean value for a property with that number of bedrooms.

5.3 Results

Hedonic Pricing Regression and Price Index – Main Results

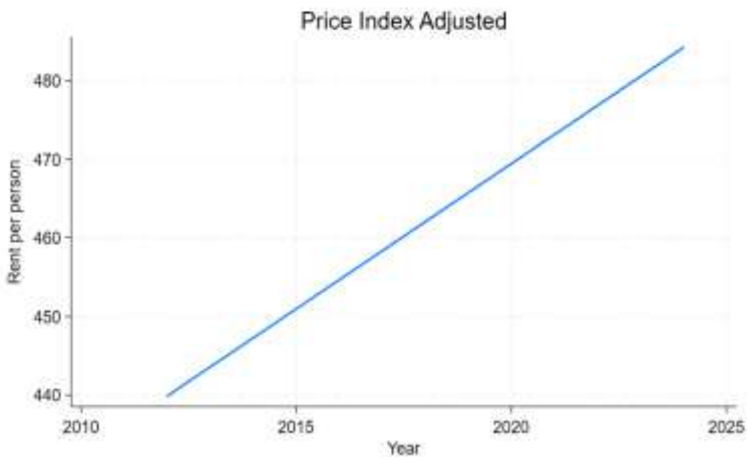
```
. reg AdjustedRentPP yearsince2012  RoomDummy* CentredDistance_weighted [aweight=Rooms
> ], noconstant
(sum of wgt is 2,004)

      Source |       SS           df       MS      Number of obs   =       845
-------------+----------------------------------   F(8, 837)       =   1821.94
       Model |   190193221         8  23774152.7   Prob > F        =    0.0000
    Residual |  10921844.7       837   13048.799   R-squared       =    0.9457
-------------+----------------------------------   Adj R-squared   =    0.9452
       Total |   201115066       845  238005.995   Root MSE        =    114.23

------------------------------------------------------------------------------
AdjustedRentPP | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
 yearsince2012 |   3.694335   1.225918     3.01   0.003     1.288099     6.10057
    RoomDummy1 |   664.9291   17.94468    37.05   0.000     629.7073     700.151
    RoomDummy2 |   484.7992    10.7281    45.19   0.000      463.742    505.8563
    RoomDummy3 |   354.5555   11.30366    31.37   0.000     332.3686    376.7423
    RoomDummy4 |   390.6292   12.69115    30.78   0.000      365.719    415.5394
    RoomDummy5 |   401.0716   14.55102    27.56   0.000     372.5108    429.6323
    RoomDummy6 |   537.0634   51.17978    10.49   0.000     436.6077    637.5192
CentredDistance_w~d |  -24.05096   3.271425    -7.35   0.000    -30.47212    -17.6298
```

Table 2: Hedonic pricing regression main results.

Our key results from the hedonic pricing regressions ("HPR") are displayed in Table 2. Principally, this result shows that monthly rent per room has increased by £3.69 (in 2012 £) each year, and this increase is statistically significant at the 1% level. The trend over time can be observed in Figure C1 above. This result is equivalent to monthly rent increasing by £44.28 over the 12-year period after adjusting for inflation. Furthermore, there is a U-shaped relationship between number of rooms and rent price per room. Rent per room is highest for 1-bed properties, which is intuitive given that these properties offer sole use of otherwise communal areas. It then declines to a minimum for 3-bed properties, and then increases again up to 6-bed properties. Note that the largest properties in this sample had six beds. The final result from the above regression suggests that rent per room declines by £24.05 (in 2012 £) for every kilometre increase in walking distance from the centre of town. This suggests that moving from the 75th percentile (3km) to the 25th percentile (1.3km) distance to town would increase monthly rent per room by £31.27. This specification was chosen as our main specification given the extremely high adjusted R-squared value of 94.5%, suggesting the model's strong explanatory power. Furthermore, the simplicity of the model alleviates concerns of over-controlling, through its use of only one distance term and one time trend. However, due to lack of data, the model incorporates few features of properties such as amenities and crucially does not provide information on the impact of HMO status.

Robustness checks and alternative specification

The regression results in Table 4 serve as a robustness check for the main specification (Table B1), with property listings in the years 2012 and 2013 excluded. As explained in the data section, the distribution of adjusted rent price per room was more dispersed and with many higher values in these years, depressing the time trend coefficient. Accordingly, it can be seen that the coefficient of years since 2014 in this specification is considerably higher (5.79) than the coefficient of years since 2012 (3.69) seen in the previous regression. Excluding these years does not have a substantial effect on any other coefficients or their statistical significance. The price index function can be seen in Figure 60, which has a steeper slope than the former.



p_year_adj1	Year
439.7792	2012
443.4735	2013
447.1678	2014
450.8622	2015
454.5565	2016
458.2509	2017
461.9452	2018
465.6395	2019
469.3339	2020
473.0282	2021
476.7225	2022
480.4168	2023
484.1112	2024

Figure 60: Price index function for HPR main specification & Table 3: Price index values.

```
.
. reg AdjustedRentPP yearsince2014 CentredDistance_weighted RoomDummy* [aweight=Rooms], nocon
> ant
(sum of wgt is 1,879)

      Source |       SS           df       MS      Number of obs   =       800
-------------+----------------------------------   F(8, 792)       =   1870.72
       Model |   180506005         8  22563250.6   Prob > F        =    0.0000
    Residual |  9552502.98       792  12061.2411   R-squared       =    0.9497
-------------+----------------------------------   Adj R-squared   =    0.9492
       Total |   190058508       800  237573.135   Root MSE        =    109.82

------------------------------------------------------------------------------------------
          AdjustedRentPP | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------------------+----------------------------------------------------------------
           yearsince2014 |   5.786236   1.342748     4.31   0.000      3.15047    8.422003
CentredDistance_weighted |  -24.12867   3.213644    -7.51   0.000    -30.43694   -17.82041
              RoomDummy1 |   659.7079   16.73716    39.42   0.000     626.8534    692.5623
              RoomDummy2 |   480.2661   9.567157    50.20   0.000     461.4862    499.0461
              RoomDummy3 |   349.4454   10.10932    34.57   0.000     329.6012    369.2897
              RoomDummy4 |   371.6765   12.51703    29.69   0.000      347.106     396.247
              RoomDummy5 |   401.7945   13.66177    29.41   0.000     374.9769     428.612
              RoomDummy6 |   592.5052   69.07968     8.58   0.000     456.9043    728.1061
------------------------------------------------------------------------------------------
```

Table ~~B3~~4: Results from robustness check, removing years 2012 and 2012.

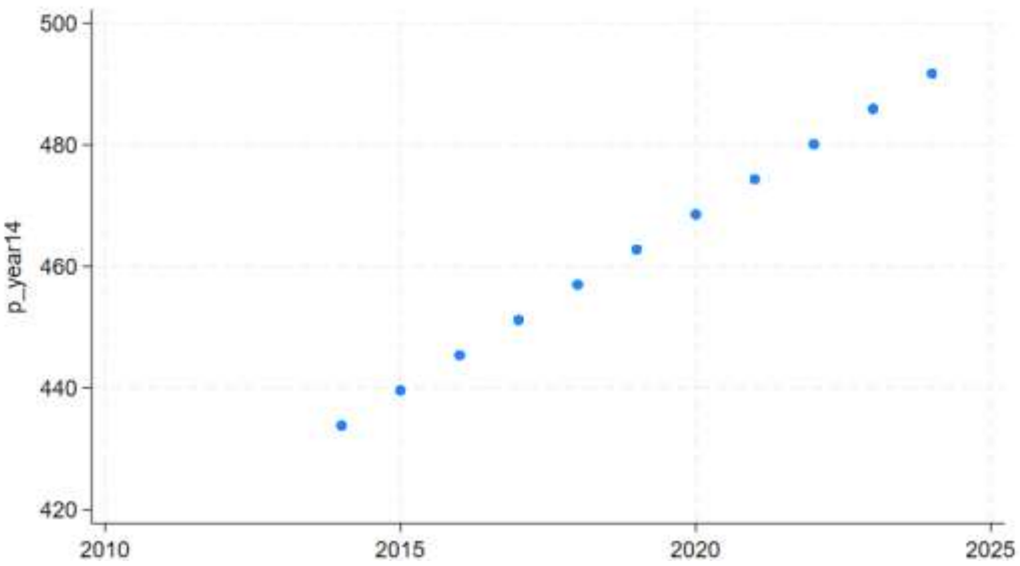


Figure 61: Price index function for HPR excluding the years 2012 and 2013.

Table 5 presents the results for an alternative regression specification which includes an interaction term between rooms and distance. The coefficient for this interaction term is not significant and its inclusion has little effect on the value of any of the other coefficients, which is why it was not included in our final specification. The result suggests that the effect of distance on rental price per room does not depend on the number of rooms in the property, and likewise the effect of number of rooms in a property on rental price per room does not depend on distance.

```
.
. reg AdjustedRentPP yearsince2012 Rooms_distance CentredDistance_weighted RoomDummy* [aweight=
> Rooms],  noconstant
(sum of wgt is 2,004)
```

Source	SS	df	MS			
				Number of obs	=	845
				F(9, 836)	=	1617.71
Model	190194142	9	21132682.5	Prob > F	=	0.0000
Residual	10920923.5	836	13063.3057	R-squared	=	0.9457
				Adj R-squared	=	0.9451
Total	201115066	845	238005.995	Root MSE	=	114.29

AdjustedRentPP	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
yearsince2012	3.734638	1.235953	3.02	0.003	1.308702	6.160575
Rooms_distance	.9293812	3.499751	0.27	0.791	-5.93995	7.798713
CentredDistance_weighted	-26.59876	10.1372	-2.62	0.009	-46.49611	-6.701416
RoomDummy1	662.5929	20.19161	32.82	0.000	622.9607	702.2251
RoomDummy2	481.3365	17.2322	27.93	0.000	447.513	515.1599
RoomDummy3	349.526	22.4402	15.58	0.000	305.4803	393.5718
RoomDummy4	383.7993	29.08028	13.20	0.000	326.7203	440.8782
RoomDummy5	393.6148	32.02205	12.29	0.000	330.7617	456.4678
RoomDummy6	531.8671	54.97736	9.67	0.000	423.9572	639.7769

Table 5: Results from alternative specification with a room-distance interaction term.

The alternative specification presented in Table 6 replaces adjusted rent per room with its natural logarithm. This allows for easy interpretation of the time trend and distance coefficients as percentages. As such, it can be said that inflation-adjusted rent per room increased 1% each year between 2012 and 2024. Furthermore, rent decreases 6% for each kilometre increase in walking distance from the centre of figuretown. Given that the room dummies are indicator variables, these coefficients cannot be easily interpreted as percentage changes.

```
. reg ln_AdjustedRentPP yearsince2012 CentredDistance_weighted RoomDummy* [aweight=Rooms], noco
> nstant
(sum of wgt is 2,004)
```

Source	SS	df	MS			
				Number of obs	=	845
				F(8, 837)	=	59718.75
Model	31369.805	8	3921.22563	Prob > F	=	0.0000
Residual	54.9587158	837	.065661548	R-squared	=	0.9983
				Adj R-squared	=	0.9982
Total	31424.7638	845	37.1890695	Root MSE	=	.25625

ln_AdjustedRentPP	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
yearsince2012	.0102525	.00275	3.73	0.000	.0048548	.0156502
CentredDistance_weighted	-.0624559	.0073385	-8.51	0.000	-.0768599	-.0480519
RoomDummy1	6.441455	.040252	160.03	0.000	6.362449	6.520462
RoomDummy2	6.136119	.0240646	254.99	0.000	6.088885	6.183353
RoomDummy3	5.829017	.0253601	229.85	0.000	5.77924	5.878794
RoomDummy4	5.89535	.0284718	207.06	0.000	5.839466	5.951235
RoomDummy5	5.949035	.0326389	182.27	0.000	5.884971	6.013099
RoomDummy6	6.238571	.1148038	54.34	0.000	6.013234	6.463908

Table ~~B5~~6: Results from log adjusted rent per room specification.

The regression used to generate the results in Table 7 was the same as the main specification but with another time trend added to control for post-pandemic effects. However, the table demonstrates that the coefficient of the 'years since 2020' variable is not significant, demonstrating that there has not been a differential time trend since 2020.

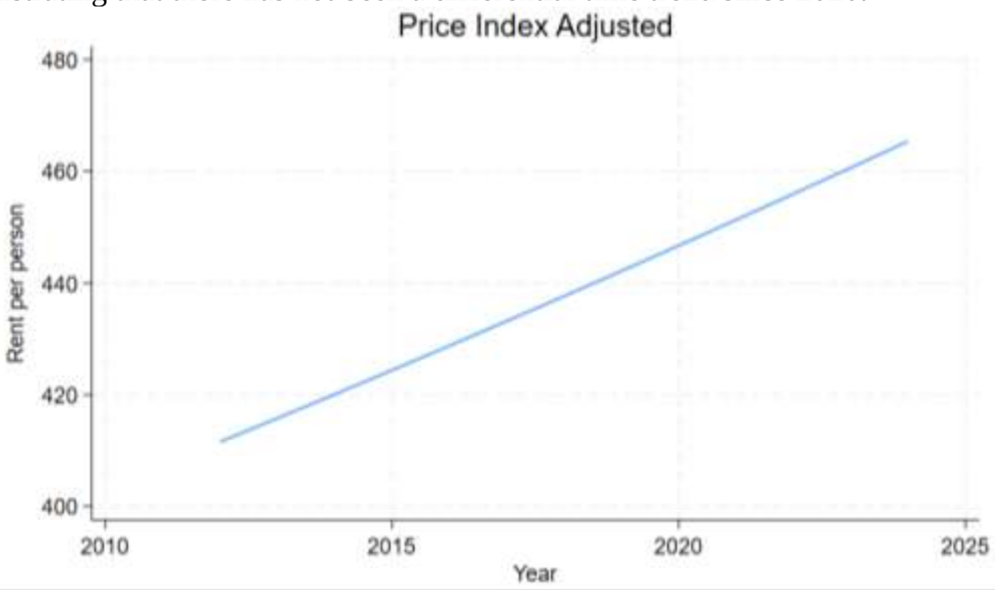


Figure 62: Price index function for log adjusted room specification.

```
. reg AdjustedRentPP yearsince2012 yearsince2020 CentredDistance_weighted RoomDummy* [aweight=R
> ooms], noconstant
(sum of wgt is 2,004)

      Source |       SS           df       MS      Number of obs   =       845
-------------+----------------------------------   F(9, 836)       =   1617.59
       Model |   190193333         9  21132592.6   Prob > F        =    0.0000
    Residual |  10921732.9       836  13064.2738   R-squared       =    0.9457
-------------+----------------------------------   Adj R-squared   =    0.9451
       Total |   201115066       845  238005.995   Root MSE        =     114.3

------------------------------------------------------------------------------------------
          AdjustedRentPP | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------------------+----------------------------------------------------------------
           yearsince2012 |   3.866454   2.227996     1.74   0.083    -.5066681    8.239577
           yearsince2020 |  -.5082519   5.492154    -0.09   0.926    -11.28828    10.27178
CentredDistance_weighted |  -24.03335   3.278895    -7.33   0.000    -30.46918   -17.59751
              RoomDummy1 |   664.3209   19.59168    33.91   0.000     625.8662    702.7755
              RoomDummy2 |   484.2056   13.19944    36.68   0.000     458.2977    510.1135
              RoomDummy3 |   353.9622   13.67165    25.89   0.000     327.1274     380.797
              RoomDummy4 |   390.0481   14.77271    26.40   0.000     361.0521    419.0441
              RoomDummy5 |   400.4344   16.68545    24.00   0.000     367.6841    433.1847
              RoomDummy6 |    536.604   51.58642    10.40   0.000     435.3499    637.8581
```

Table ~~B6~~7: Results from the post-pandemic time trend specification

```
. *Running main regression with distance to library added
. reg AdjustedRentPP yearsince2012  RoomDummy* CentredDistance_weighted CentredDistanc
> egym_weighted [aweight=Rooms], noconstant
(sum of wgt is 1,471)

      Source |       SS           df       MS      Number of obs   =       621
-------------+----------------------------------   F(9, 612)       =   1551.18
       Model |   141742911         9  15749212.3   Prob > F        =    0.0000
    Residual |  6213651.33       612   10153.025   R-squared       =    0.9580
-------------+----------------------------------   Adj R-squared   =    0.9574
       Total |   147956562       621  238255.333   Root MSE        =    100.76

------------------------------------------------------------------------------
AdjustedRentPP | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
---------------+--------------------------------------------------------------
 yearsince2012 |   3.023688    1.28523     2.35   0.019     .4996909    5.547684
    RoomDummy1 |   618.4776   19.69741    31.40   0.000     579.7949    657.1603
    RoomDummy2 |   447.4881   12.14694    36.84   0.000     423.6334    471.3429
    RoomDummy3 |   346.1887   12.02762    28.78   0.000     322.5683    369.8091
    RoomDummy4 |   366.1953   13.91941    26.31   0.000     338.8597    393.5309
    RoomDummy5 |   365.6016   15.84991    23.07   0.000     334.4748    396.7284
    RoomDummy6 |   478.5945   45.86927    10.43   0.000     388.5142    568.6748
CentredDistance_w~d |  -61.44991   6.609881    -9.30   0.000    -74.43071   -48.46911
CentredDistancegy~d |  -17.83685   6.884117    -2.59   0.010    -31.35621   -4.317495
```

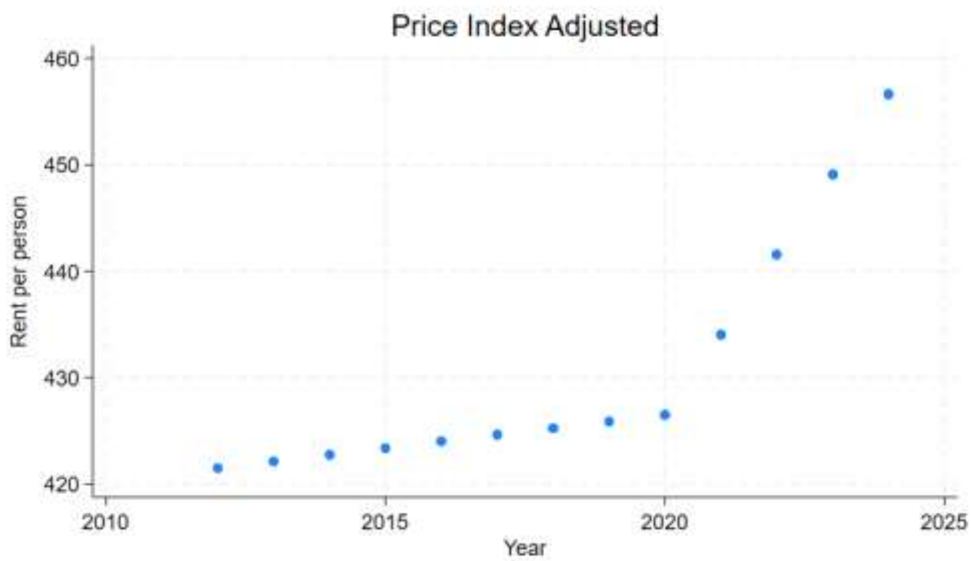
Table ~~B7~~8: Results from distance from the University Sports Centre/Gym specification.

The results in Table 8 show that, holding all else constant (including distance from the town centre), each kilometre increase in distance from the University Sports Centre leads to a £17.84 reduction in adjusted monthly rental price per room, however this result is only marginally significant (10%). Furthermore, this result increases the magnitude of the coefficient of distance from the centre of town, suggesting that perhaps some properties' relative distance from town is compensated for by their relative proximity to the gym, limiting the reduction in rental price. This proximity measure may be unrelated to the gym itself, and may simply reflect that areas closer to the gym are considered better areas to live (e.g. nicer-looking streets, generally more appealing properties, more nearby amenities, closer to the science building – the only university buildings not in the centre of town, etc). \

```
. reg AdjustedRentPP yearsince2012 yearsince2020 RoomDummy* CentredDistance_weighted C
> entredDistancegym_weighted  [aweight=Rooms], noconstant
(sum of wgt is 1,471)

      Source |       SS           df       MS      Number of obs   =       621
-------------+----------------------------------   F(10, 611)      =   1397.13
       Model |   141757150        10    14175715   Prob > F        =    0.0000
    Residual |   6199411.9       611  10146.337    R-squared       =    0.9581
-------------+----------------------------------   Adj R-squared   =    0.9574
       Total |   147956562       621  238255.333   Root MSE        =    100.73

------------------------------------------------------------------------------
AdjustedRentPP | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
 yearsince2012 |   .6253891   2.397751     0.26   0.794    -4.083445    5.334223
 yearsince2020 |   6.903071   5.827075     1.18   0.237    -4.540455     18.3466
    RoomDummy1 |   628.5833   21.45931    29.29   0.000     586.4403    670.7263
    RoomDummy2 |   456.8601   14.49264    31.52   0.000     428.3986    485.3215
    RoomDummy3 |    355.958   14.57989    24.41   0.000     327.3252    384.5907
    RoomDummy4 |     376.62   16.46386    22.88   0.000     344.2874    408.9526
    RoomDummy5 |   375.3997   17.87345    21.00   0.000     340.2988    410.5005
    RoomDummy6 |   485.5766   46.23138    10.50   0.000     394.7849    576.3682
CentredDistance_w~d |  -62.48749   6.665497    -9.37   0.000    -75.57755   -49.39742
CentredDistancegy~d |  -17.51185   6.887316    -2.54   0.011    -31.03753   -3.986162
------------------------------------------------------------------------------
```

Table 9: Results from combined post-2020 time trend and distance from the University Sports Centre/Gym specification.

Table 9 displays the results for the main regression with controls added for post-pandemic time trend and distance from the University gym. In this specification we appear to be over-controlling, as both time trends are now highly insignificant. The price index function for this regression is shown in Figure 63, showing a definitive kink at the year 2020, reflecting the coefficient of the variable 'years since 2020' being more than ten times that of the coefficient of 'years since 2012'. Note, however, that neither are statistically significant.



~~Figure C4: Price index function for combined post-2020 time trend and distance from the University Sports Centre/Gym specification.~~Figure 63: Price index function for

Table 10 presents results from the initial regression with a term added to control for the concentration of HMOs in the property's postcode. Interpreting the coefficient suggests that a 10% increase in the probability that a property is an HMO increases the monthly rental price of a room by £18.81. While this appears to be highly significant, there is concern of reverse causality or omitted variable bias from the introduction of this coefficient. In particular, it is likely that 'higher-value' properties are selected to be HMOs as landlords consider it worthwhile to pay the administration fees, or that certain areas of town have a higher concentration of HMOs as they are known to be more appealing to students and so command a higher price.

```
. reg AdjustedRentPP yearsince2012  RoomDummy* CentredDistance_weighted HMO_to_Propert
> y_Ratio_ [aweight=Rooms], noconstant
(sum of wgt is 1,425)

      Source |       SS           df       MS      Number of obs   =       603
-------------+----------------------------------   F(9, 594)       =   1555.11
       Model |   137886888         9  15320765.3   Prob > F        =    0.0000
    Residual |  5852008.24       594  9851.86572   R-squared       =    0.9593
-------------+----------------------------------   Adj R-squared   =    0.9587
       Total |   143738896       603  238372.962   Root MSE        =    99.257

------------------------------------------------------------------------------
AdjustedRentPP | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
 yearsince2012 |   3.672915    1.30851     2.81   0.005     1.103046    6.242785
   RoomDummy1 |   597.4338   19.18153    31.15   0.000      559.762    635.1057
   RoomDummy2 |   430.0718   11.86984    36.23   0.000     406.7598    453.3837
   RoomDummy3 |   323.4187   12.17119    26.57   0.000     299.5149    347.3225
   RoomDummy4 |   350.3278   14.11499    24.82   0.000     322.6064    378.0491
   RoomDummy5 |   336.9172   16.25357    20.73   0.000     304.9957    368.8386
   RoomDummy6 |   398.6554   47.10227     8.46   0.000     306.1481    491.1626
CentredDistance_w~d | -54.34814   6.652715    -8.17   0.000    -67.41384   -41.28243
HMO_to_Property_R~_ |  188.1328   36.43057     5.16   0.000     116.5844    259.6812
------------------------------------------------------------------------------
```

Table ~~B9~~10: Results from specification including ratio of number of HMOs to total properties in a postcode.

Fife Council Rent Data Analysis

```
regress RentPP2012 twobedroom threebedroom fourbedroom fivebedroom CuparNW Dunfermli
> neWF KirkcaldyCF nonStA_StAHMA DetachedHouse SemidetachedHouse Flat Furnished PartFu
> rnished RentPPdoublemean if NoofBedrooms !=0 & PropertyType != "Other"

      Source |       SS           df       MS      Number of obs   =       626
-------------+----------------------------------   F(14, 611)      =    142.04
       Model |  11193683.3        14  799548.808   Prob > F        =    0.0000
    Residual |  3439408.92       611  5629.14717   R-squared       =    0.7650
-------------+----------------------------------   Adj R-squared   =    0.7596
       Total |  14633092.2       625  23412.9476   Root MSE        =    75.028

------------------------------------------------------------------------------
  RentPP2012 | Coefficient  Std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
  twobedroom |  -174.9379   7.852674   -22.28   0.000    -190.3594   -159.5164
threebedroom |  -221.4803   10.01382   -22.12   0.000     -241.146   -201.8146
 fourbedroom |  -219.5001   15.73265   -13.95   0.000    -250.3967   -188.6035
 fivebedroom |  -234.7074   31.47515    -7.46   0.000      -296.52   -172.8948
     CuparNW |  -147.1517    15.8244    -9.30   0.000    -178.2285   -116.0748
DunfermlineWF |  -132.3786   12.79793   -10.34   0.000    -157.5119   -107.2453
 KirkcaldyCF |  -160.6217   12.52613   -12.82   0.000    -185.2212   -136.0223
nonStA_StAHMA |   -95.9548    13.0698    -7.34   0.000     -121.622   -70.28762
DetachedHouse |   33.90469   13.13835     2.58   0.010     8.102879     59.7065
emidetachedHouse |   11.2303   12.92343     0.87   0.385    -14.14944    36.61004
        Flat |  -21.06448   9.370071    -2.25   0.025    -39.46593   -2.663027
   Furnished |   57.65448   9.280315     6.21   0.000     39.41751    75.89144
PartFurnished |   30.66204   13.60398     2.25   0.025       3.9458    57.37827
RentPPdoublemean |  357.5487   19.42723    18.40   0.000     319.3964    395.7009
       _cons |   575.6479   15.66868    36.74   0.000      544.877    606.4189
------------------------------------------------------------------------------
```

Table ~~B10~~11: Results from regression using Fife Council's rent data.

Table 11 shows the results from regression of rent per room on various property characteristics using the dataset provided by Fife Council. The results from this dataset reflect our main specification in many ways, including properties being further away commanding lower rents. This regression also suggests some new results based on the availability of data on property characteristics which are not represented in our main dataset, including whether a property is a flat or house, its furnished status, as well as data on properties outside of St Andrews but still within Fife. While these findings may be useful in informing our understanding of the determinants of rent prices in St Andrews, it is important not to place too much weight on them given the uncertainty around data collection methods, hence our decision not to base our analysis on these results.

5.4 Discussion

Analysis of results

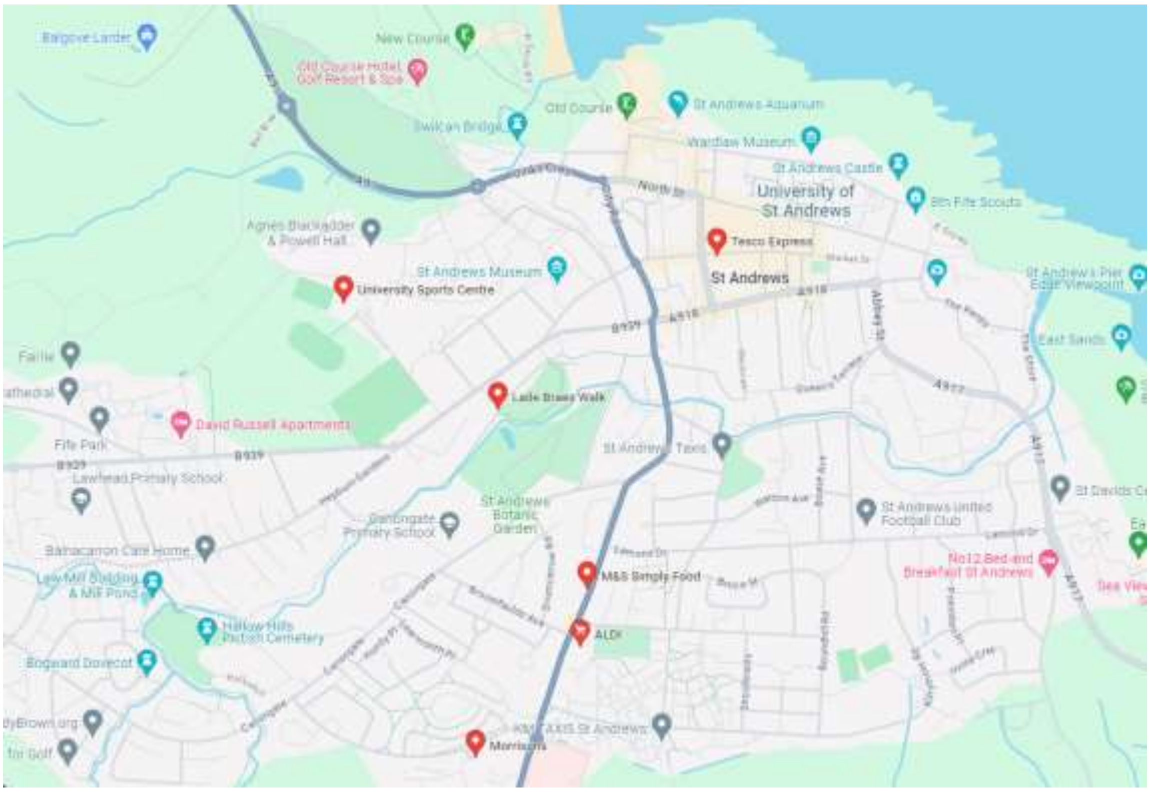
The results from our main specification, displayed in Table 10 above, suggest a persistent increase in inflation-adjusted rent year on year. There are many plausible explanations for this. The past decade has been marked by nation-wide rent price increases, suggesting factors external to the St Andrews context may be the cause. Alternatively, rent prices may have increased due to a demand shock from exogenous increases in students moving into St Andrews. Alternatively, there may have been supply contractions, perhaps in anticipation of the new HMO regulations implemented in 2019, or due to the trend of landlords leasing out their properties for short-lets (not included in our dataset), as long-lets command lower prices.

The finding that there is a premium for renting properties closer to the centre of town in St Andrews is reflective of student sentiment of being more than 10 minutes from town being fairly far away. A similar sentiment can be extrapolated to students' reluctance to live outside of town, such as in neighbouring towns or in Dundee, a 30-minute bus journey away. The final result was that there is a U-shaped relationship between rent price and number of rooms in a property. The decline in rent price down to 3-bed properties perhaps reflects having to share communal spaces with more users. The increase after the 3-bed threshold may reflect that disproportionately more of the sampled properties are HMOs (as families renting are unlikely to require very large properties), for which landlords can charger disproportionately high rents given that there are more rent-paying occupants than in a typical family.

The alternative specifications also give useful insight into the determinants of rent prices in St Andrews. The specification including the rom-distance interaction term (Table 5) showed that the coefficient of this term was not statistically significant. This suggests that the effect of total number of rooms in a property does not depend on the property's distance from the centre of town. This is an informative result, as would have been plausible for this interaction to be significant, for example if students preferred to live in bigger properties if the properties were further away to avoid feeling isolated, or if some students had a preference both to live further from town and to live with fewer people.

The log-adjusted rent per room specification in Table 6 showed that increases in rental price may be exponential rather than linear. This finding is relevant for policymakers, as affordability of rental prices in Fife is already an issue of public concern (Fife Council, 2024), and an exponential rather than linear increase could mean that the situation could become out of hand relatively more quickly than the main results suggest.

The specification which includes the post-2020 time trend (results in Table 7) showed that the coefficient of the 'years since 2020' variable was not significant. This result is interesting as the pandemic had significant disruptive effects on the rental market nation-wide and is also cited as a time when the University began admitting increasingly large cohorts of students. The insignificance suggests that these and other pandemic-related changes did not have a significant medium-term effect on rental prices.



~~igure C5: Map of St Andrews, displaying the many amenities in the West.~~ Figure 64: Map of St Andrews, displaying the many amenities in the West.
Source: Google Maps.

The specification including the university gym and sports centre (results in Table 8) is useful in understanding the importance of amenities other than the centre of town in determining rental price. However, the coefficient of weighted distance from the university gym is only marginally significant when controlling for weighted distance from the centre of town. Also, this coefficient could reflect many amenities in the Western part of St Andrews, such as the University science buildings, Lade Braes public footpath, or the trio of Aldi, Morrisons and M&S within 100m of each other on Largo Road (see Figure 64). As such, while this result is of interest in that it suggests that the centre of town is not the only amenity impacting property price, interested parties may be interested in exactly which amenities raise rental prices and by how much.

Finally, the specification including the ratio of number of HMOs to total properties (results in Table 10) is a useful starting point in understanding what the impact could be of HMO licence status of a property on its rental price. However, the results presented cannot credibly be argued to be causal, given the concerns relating to reverse causality and omitted variable bias. Reverse causality is plausible if properties landlords are more willing to pay to HMO administration fees for properties that command higher prices, and omitted variable bias is plausible if other characteristics (like the properties having certain amenities like a disproportionate ratio of bathrooms to bedrooms) may cause both higher rental prices and the increased suitability of a property as an HMO rather than a family dwelling. These endogeneity issues could be resolved through an analysis using a similar method to that off Duflo & Pande (2007). This would require data on total number of HMOs by postcode in St Andrews for all years from 2012-2024, as well as “HMO incidence” of postcodes (ratio of total number of HMOs to total number of properties) in 2012. One could then create an instrument for predicted HMO incidence in all following years by assuming that the attribution of HMOs to postcodes each year matches the HMO incidence in 2012, and regressing rental price on this instrument in a 2SLS regression. However, due to lack of data on HMO-to-total properties ratio in 2012, this analysis could not be carried out in this report.

Opportunities for further research

The addition of further data on property amenities would be useful in improving the model. Relevant data would include whether or not the property has a garden, how many bathrooms it has or its bathroom-to-bedroom ratio, whether the property is a flat or a house and whether it is being leased as an HMO or not. The latter detail would be particularly relevant to assessing the impact of the HMO Overprovision Policy on rental price, which could be done using a regression discontinuity design. In the absence of data on HMO status, HMO-to-property ratio by postcode could be a useful proxy as mentioned above, and perhaps entails more easily attainable data.

Finally, the recent unprecedented increases in interest rates may have an impact on the cost of rental prices in St Andrews. The collection of more data and over a longer period of time will be necessary to assess the effects of these interest rate hikes in the medium- to longer-term.

6. Demand Team

The demand team was focused on building a model of the rental market that could be calibrated using observed prices and supply numbers, and then used to perform counterfactual analyses to answer questions about the effects of supply and demand shocks on rent in St Andrews. Our main objective was to separate the effects of the supply shock of the capping of HMO licenses and the demand shock of the sharp increase in student population. We did this to determine which factors, demand, or supply, contributed more to the overall rise in rent observed after 2019. Additionally, we were interested in counterfactual questions, such as the hypothetical scenario of the impact that the building of another student hall would have on rent in the private market. A large motivator of our team's research was the policy question of, "If Fife Council decided to allow for more HMO licenses in the market, by how much would they have to raise the cap to ease the recent rise in rent?" We used the model to answer these questions using our developed model and rental market observations, the results of which are discussed below.

The members of the demand team are Georgi Butch, Ajda Vili Kapelj Zupan, and Marcin Weremczuk.

6.1 Methods

We model average monthly rent per bedroom in the housing market of St Andrews. We built our model in Python, by constructing a system of several market-related equations, which take input data for all corresponding unknowns and can be used to solve for the chosen investigated one (rent) by inputting observed data.

Below are the equations we included in our model. For each year, we have a corresponding set of equations for each of the ones below.

We are representing the quantity of rooms demand as a linear function of price:

$$D(p) = A - B \cdot p$$

where D(p) stands for quantity of rooms demanded and p for price, with A and B being constants representing the y-intercept and slope of the demand curve, respectively.

Below is the standard equation for point elasticity at equilibrium price.

$$\varepsilon = \frac{d\big(D(p)\big)}{dp} \cdot \frac{p}{D(p)}$$

Where ε stands for demand elasticity, and D(p) is the equilibrium quantity of rooms supplied at the equilibrium price, which our model assumes is the same as the HMO provision for that year, under the following market condition:

$$D(p) = S$$

where S strands for supply, or the HMO provision of rooms.

Below are the equations for our assumption of demand shock, which we have a set for every year that we considered when doing counterfactual analysis. Here we are using parameters of a respective considered year and current year (2023):

$$ratio = \frac{S_y}{S_c}$$
$$D_y(p_c) = ratio \cdot D_c(p_c)$$

Above y stands for the considered year and c for the current year (2023).

When investigating decomposition and doing other counterfactual analysis we are inputting observed data for demand (D), supply (S) and rent (p). The coefficients A and B are simultaneously calculated by solving the above system of equations.

In this work we are assuming perfectly inelastic supply of HMO properties. We are calibrating for demand elasticity by using observed data for the parameters stated above. Our demand elasticity value is the same for each considered year, and we are determining this value by a calibration process which is included in our model. This process considers a range or elasticity values and minimizes the difference between the observed and modelled rent paths. The corresponding elasticity (that minimizes this difference) is set as the input elasticity parameter of the model. Our calibrated elasticity value is 1.4.

Below is a graph of the observed rent path and the modelled rent path when their difference is minimized – when price elasticity of demand is 1.4 (for all years in the 2016-2023 period).

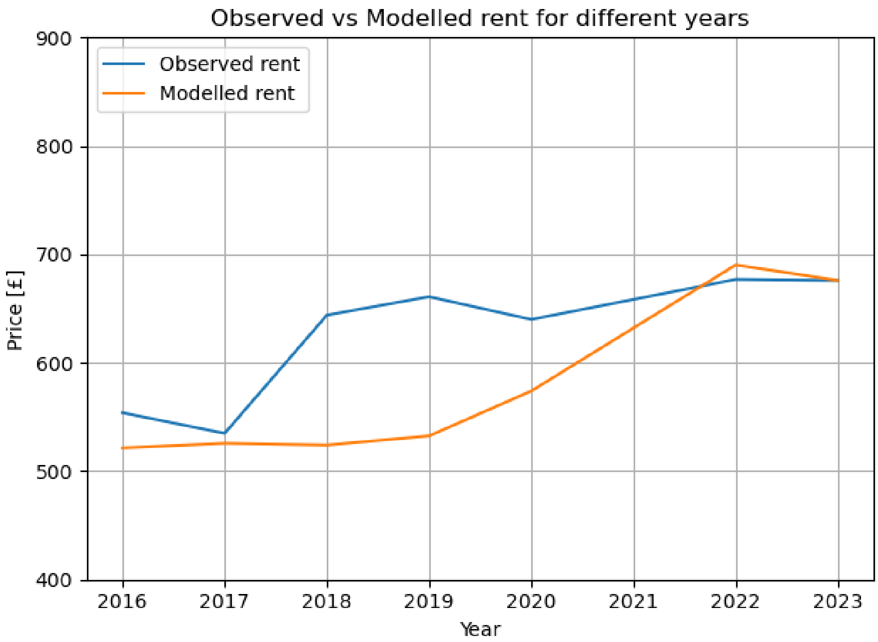


Fig. D1: Observed vs Modelled rent path for years 2016-2023

When performing counterfactuals to predict a future private rent market equilibrium, we add an additional set of equations for the future year and substitute the supply and demand inputs as desired to predict a figure for future rent.

The full codes can be found in the appendix.

<u>Assumptions</u>

While constructing our model and method of decomposition and counterfactual analysis, we have made assumptions related to demand and supply of this market. The following have the main implications in our method:

- Supply is perfectly inelastic (determined by the built housing).
- Active HMO licenses determine how many effective rooms there are available – hence determine the supply.
- Demand of this housing market is a linear function. Only students in St Andrews create demand on this market.
- Students live only in St Andrews and in HMO licensed properties.
- Students have a willingness to pay for a room in St Andrews relative to the alternative (living outside town), creating demand with some elasticity in the town's housing market

6.2 Data

The data we used for prices consists of the price index constructed by the Price Team earlier in the semester, which was a weighted average of per-bedroom rents from 2016 to 2023. For supply, we averaged the number of HMO rooms available across 12 months for each year from 2016 to 2019, using the data from the Register Team. These numbers exclude the HMO licensed rooms in student halls, so that we can isolate the supply on the private market. For all years including and after 2019, when the HMO policy went into effect, we use a constant supply of 2718 rooms, which is the total number of rooms available under the current HMO provision.

For demand, we used the University of St Andrews's data on the number of students living in halls found via a Freedom of Information Act request. We subtracted the number of students living in halls from the total number of students enrolled in each year to find the number of students that do not live in university-provided accommodation in each year. These students constitute the demand for housing in the private market for each year.

As mentioned in the methods section, our value of demand elasticity, 1.4, was found through calibrating the the model, rather than being taken from previous studies, which typically assume that the demand for housing is inelastic, or less than 1. When performing counterfactual analysis, we assumed that this elasticity was constant across all years, including in 2024 for future scenarios.

YEAR	DEMAND	SUPPLY	MONTHLY RENT
2016	5003	2764	554
2017	5343	2879	535
2018	5071	2758	644
2019	5175	2718	661
2020	6381	2718	640
2022	6031	2718	677
2023	6000	2718	676

Table D2: Data inputs for the linear demand and supply model

The above table shows all data inputs for the model. These values are taken from our observations of the market. It should be noted that the quantity demanded figure for 2023 of 6000 is an estimate based on the assumption that the number of students searching for housing in the private market stayed constant from 2022 and might have even slightly decreased due to lower enrolment of the university. The numbers of students in halls for this year were not yet available. All prices have been adjusted for inflation (CPI) using 2023 as the base year.

6.3 Results

Decomposition of the observed rent increase into supply and demand shocks

Our main topic of interest was to what extent the recent rent increase was caused by the HMO Overprovision policy (implemented in 2019) and to what extent by the growing student population. We investigated this by a counterfactual analysis, decomposing the market into two isolated scenarios: the case where the policy is not implemented and the case where student population does not grow since the policy was implemented. We call these two cases "no supply shock" and "no demand shock" respectively. To investigate "no supply shock" we assumed that yearly supply increases by 1.5%, since this was an average rate of change in supply over 5 years before the policy implementation. For investigating "no demand shock" we assumed that the student population stayed constant from the policy implementation on.

Hence, we determined that 29% of the rent increase could be attributed to the HMO Overprovision policy, while 71% was linked to the rise in student numbers. In total, the modelled rent per room increase from 2019 to 2023 was found to be 9.84%, from £569 to £625 per month. The decomposition results are demonstrated graphically on Fig. D3 below.

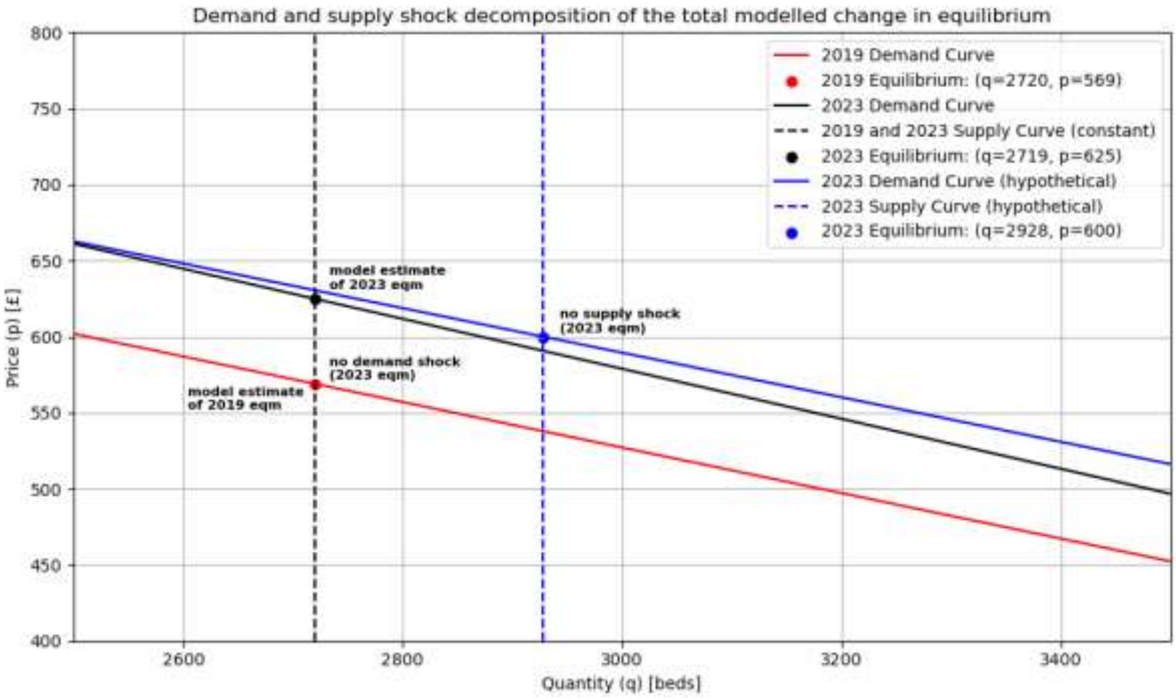


Fig. D3: Decomposition of the total change in the modelled market equilibrium into demand and supply shocks

Hypothetical construction of a new student hall

Another question we answered with our model concerned the impact of the hypothetical construction of another student hall by the University. We found that the median size of an undergraduate hall, excluding the outlier of David Russell Apartments, which is a large student residential complex, was 200 beds. Knowing this, we ran the model with the figure for students not in halls set to 5800 in 2024, representing 200 students leaving the private rental market and choosing to live in the new hall. Our model predicts that this would lower the equilibrium rent per room by 2.4%, from 676 in 2023 to 660 in 2024. This also assumes that the university would build the hall but not significantly increase the number of enrolled students. We show these findings on Fig. D4 below.

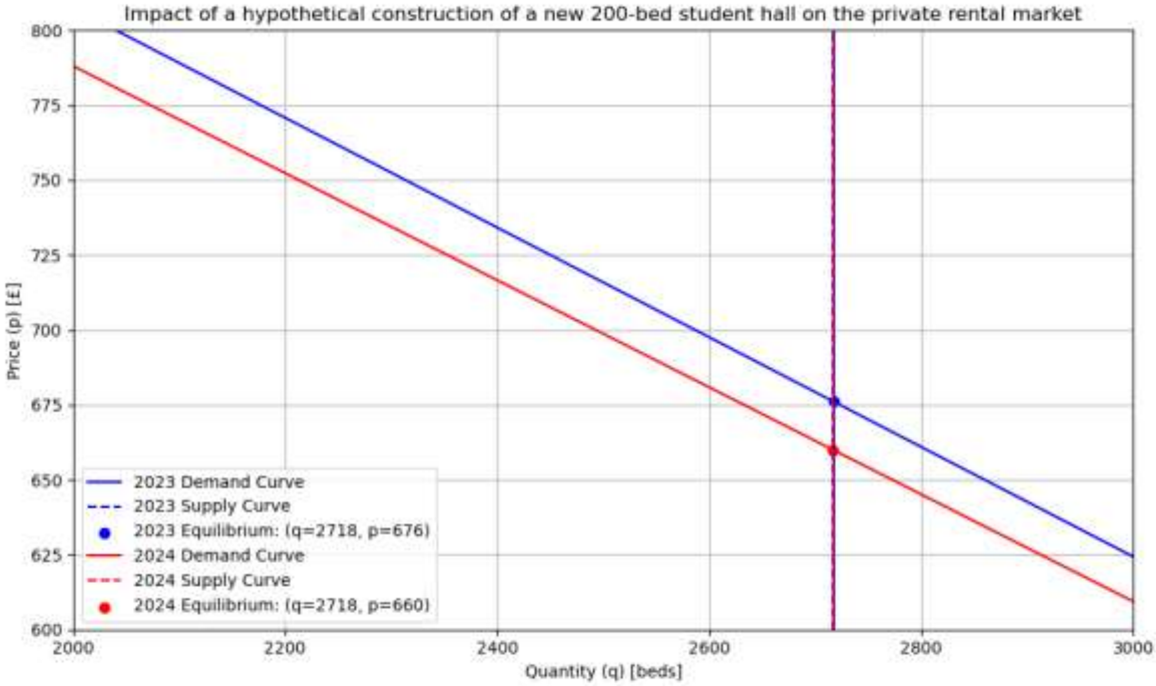


Fig. D4: Projected impact of a hypothetical construction of a new 200-bed student hall

<u>Required change in supply to decrease rent by 5%</u>

If we are to assume that the HMO policy had a significant effect on the recent increase in rent in St Andrews, which our model suggests it did, then we can also assume that increasing or decreasing the HMO provision levels may be used as a reasonable method of adjusting the market rent. It may be desirable to bring the average rent back down to pre-policy levels, such as those observed in 2018. To do this, a 5% decrease in equilibrium rent is required, from 676 to 644. Our model predicts that the number of HMO rooms available would have to increase by 7.5%, from 2718 beds to 2922, to bring about such a change. This is shown on Fig. D5 below.

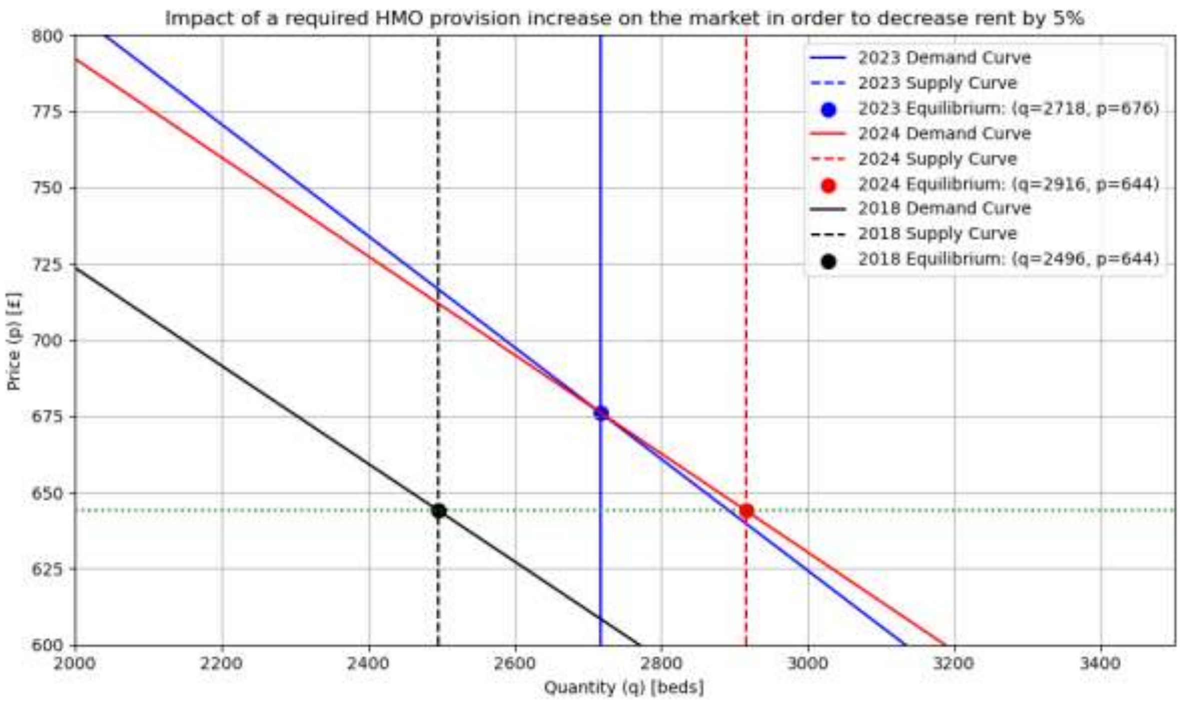


Fig.D5: Impact of a hypothetical increase in HMO licence provision required to decrease rent by 5%

<u>Continued increase in student numbers into the future</u>

One final counterfactual we investigated was rooted in the growth patterns of the student population over recent years. We find that the average increase in student population for each year is about 5%. If this continues without any changes from Fife Council in the HMO provision, or any increase in university-provided accommodation, and if we assume that the change in students looking for private housing is proportional to the overall change in enrolment, we can predict the increase in price by inputting a demand figure of 6300 students. The model returns a predicted price increase from 676 to 702, which is about 4% higher. This is shown on Fig. D6 below.

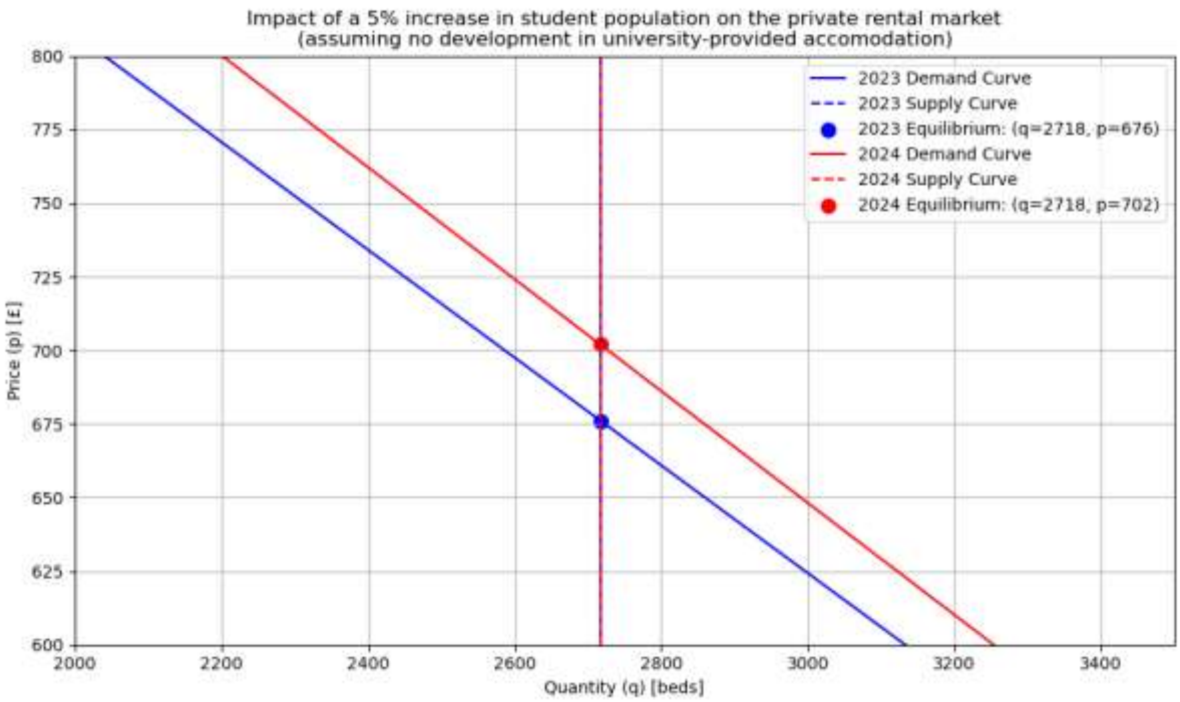


Fig. D6: Projected impact of a 5% increase (relative to 2023) in student population on the private rental market, assuming no growth in university-provided accommodation

6.4 Discussion

Decomposition findings

Since we found the demand on the private rental market in St Andrews to be elastic, it could have been expected that we determined that demand factors had a relatively larger impact on rent levels than supply factors. Most of recent years' rent increase was attributed to growing student population due to expansion of the university. However, limiting HMO licence provision still considerably influenced the market outcome, contributing to almost 30% of the total rent increase. It may be concluded that adjusting both demand and supply can effectively introduce relevant changes on the market and influence rent levels.

Implications of model assumptions

The design of our model is based on a few core assumptions, which can be elaborated on and further tested in the future.

First, we assume that demand is a linear function consisting of only the students of St Andrews, including undergraduates, postgraduates, and people of varying fee statuses and other personal characteristics. This is a reasonable assumption for the demand for HMO properties because it is likely that students are the ones who are most likely searching for properties which require HMO licenses – those which house three or more unrelated people.

We also assume that supply is perfectly inelastic (supply curve is vertical), at the number of rooms HMO rooms available during any given year. This is an oversimplification, but also a reasonable one.

The third assumption we make is that all students not living in halls are demanding an HMO property. This does not account for two significant groups: commuter students and students who seek housing on the private market but end up living in a one or two bed property that does not require an HMO license. Investigating these groups further could prove to be a valuable way of improving the accuracy of the model's predictions.

We also assume that the input data is reliable and precise, which might not be the case. The weighted average price index, which was used for the price observations input, was based on all logged properties within St Andrews limits, whether they had an HMO licence or not. Our model, however, only is concerned with the market for HMO properties, therefore these observations might not be reflective of the rent levels for HMO properties only. Furthermore, issues with the HMO register may result in an inaccurate determination of supply on the market.

Introduced model improvements

For this progress report, we have decided to use a different price index than what we had previously used as input data for the model. We had used the yearly prices from Ayton House but have since started using the weighted-average based price index collected through scraped data from the Price Team. We believe this gives us a more accurate overall snapshot of rent changes from year to year across the market, rather than focusing on the rent of just one establishment.

We have also updated our elasticity assumption based on the calibration techniques mentioned above using the new price index. We had previously assumed that demand for housing in St Andrews was inelastic, at a value of 0.7. However, due to the nature of a small student town with many nearby substitutes, a more elastic figure of 1.4 can be justified, as students can fairly easily move to nearby towns, where rent tends to be relatively lower as well.

Calibration and possible overfitting

It should be noted that our calibration method for the model and counterfactual analysis introduces potential overfitting, impacting the estimated demand elasticity and results. Namely, our model is constructed so that whenever we wish to model specific rents (i.e., use the system of equations to solve for these rents), we are required to leave input observed rent data for at least one year. This results in overfitting the modelled rents to that observed rent input. When calibrating the model, we used the current rent (2023) for modelling the rest, which resulted in overfitting the modelled price path on this current rent. This overfitting may have influenced our elasticity assumption, so further analysis into this estimation could be done to improve it. When conducting decomposition analysis, we used rent from 2016 to model the rest, as we wished to investigate in what rent increase (between years 2019 and 2023) a hypothetical no policy implementation and constant demand would result. That is, we did not want to overfit the modelled price for 2023, as this was our point of investigation. However, it is reasonable to assume that our decomposition analysis results might have been influenced by overfitting the modelled rents on rent from 2016. Furthermore, when conducting the rest of the counterfactual analysis, we used rent from 2023 to model the remaining rents, so we assume results of this analysis might as well be influenced by overfitting the modelled rents on rent from 2023.

It should also be noted that we have removed 2021 from the model. This is because this year is an outlier due to the conditions created by the COVID-19 pandemic, when an above average number of students were commuting.

Possible improvements

Additionally, further work can be done to improve the model. Including more previous years could improve accuracy, although this would likely require the reforming of the model design, perhaps by using for-loops or list comprehensions to make the code more user-friendly for counterfactual analysis.

Another potential avenue would be to split the demand curve by student groups, perhaps by fee status or between undergraduate and postgraduate students. These groups likely have different elasticities and willingness to pay, making it more accurate to view them separately.

Improving the accuracy of data collection for model inputs would also improve the significance of the model output. Using a price index basing only on HMO licenced properties would fit the assumptions of the model better, and hence the model would be more reflective of reality. Introducing separate models for the markets for non-HMO properties and for HMO-licenced properties would also enable us to investigate the levels of a potential HMO rent premium in St Andrews.

7. General Lessons

With this year's team structure being significantly different to those of previous years, the VIP project has experienced a major transformation this semester, having (i) a larger team overall, (ii) a team re-restructuring mid-way through the semester, and (iii) a postgraduate student being part of the project. Whilst this has opened up many new opportunities for the progress of this project, it has also led to general lessons being learnt within each team individually and the team as a whole.

Communication Dynamics and Collaborative Interactions

The vitality of robust communication mechanisms cannot be overstated. Having learnt from last year's development of the project, communication this year has been improved to flow through three steps, thereby ensuring more effective team dynamics and interactions.

Firstly, there is team-internal interaction within each of the sub-teams of the project which is crucial, as the individual team's outputs drive the overall progress of the project. However, having several team members requires everyone to (i) be flexible enough, so that schedules can be aligned, (ii) communicate problems openly and early, so that other team members can provide any help needed, and (iii) work together in setting interim deadlines and sharing results, so that the individual team work can be brought into the plenum and be used by other teams. Individual teams have learned that proper file management, meeting documentation, and collaboration is key for a good research output (more in paragraph about Effective Methods, Workflow, and Organisation).

Secondly, knowing from previous years that each sub-team has difficulties in terms of sorting their results and contribution into the overall picture of the project, the new role of a "flexible" has been introduced, who serves as the connection between each individual team and is part of the project to monitor, manage, and inform everyone to ensure a long-term conference-oriented and academic focus from a student view. Especially, as a postgraduate and having a broader view onto the project and topic, the main goal is to make sure the work is answering relevant economic questions, however, this role is dependent on the results of the other teams' work, which is why the main lesson learned has been that there is the need for an open exchange of ideas and further, early discussion with all teams, which has worked well for goal-setting and for improving and extending existing work.

Thirdly, communication with supervisors is paramount for the success of any project and has also gained importance in this project. By maintaining open lines of communication, teams are able to benefit from timely feedback, clarifications on project objectives, and insights into best practices. In particular, regular team and project-level meetings in form of Weekly Supervised Meetings (WSM) serve as pivotal forums for aligning objectives, clarifying expectations, and fostering a cohesive team environment.

An exemplary instance of overall effective flow of communication through the three steps are this year's implemented weekly Weekly Unsupervised Meetings (WUM) which ensure that feedback from peers and supervisors is well received and which serve as an open platform to exchange progress, as often supervisors stay for a longer chat and input as well.

Effective Methods, Workflow, and Organisation

A project's success does not rely only effective communication, but also lies in the implementation of effective methods, streamlined workflow, and meticulous organisation. With all sub-teams having (i) regular meetings amongst themselves and minuting these (see Survey Team for example), next to (ii) having their own sub-channel on Teams to communicate amongst themselves, and (iii) having their own folder within the project (see Register Team for example) which serves to properly structure code repositories and the team's files, confusion as to where to find files or previous discussions has been reduced.

Academic Research, Analysis, and Skills

Ultimately, the project’s main lesson is the profound value found in interdisciplinary collaboration and methodological rigor through the project offering a ground for all participants to try themselves in an academic environment without pressure. Through collective efforts of this semester’s team, new changes in the housing market could be explored with the project thereby fostering the cultivation of critical thinking, problem-solving, and research skills (including quantitative skills such as Stata, R, Python, or qualitative skills such as survey writing, academic writing, or setting up research hypothesis for research) amongst team members.

Process of Learning

The process of learning within the VIP project was characterised by experimentation, adaptation, and resilience. While some initiatives failed to yield immediate results, they provided valuable insights into the importance of clear communication, structured workflow, and collaborative problem-solving.

One significant challenge revolved around adapting to the larger team size and the subsequent need for more structured communication channels. As initially last semester’s team had a micromanagement approach to the project, that is, each team worked within their own team and did not set a focus on interdisciplinary communication, a more formalised approach to work and communication was established this semester which included the introduction of a "flexible" role to bridge communication gaps between sub-teams. This role proved effective in facilitating cross-team collaboration and ensuring alignment with project goals, however, defining the responsibilities and scope of this role led to difficulties in its execution. Through iterative discussions and adjustments, the “flexible” role was refined to not only act as a link across teams and create an overall academic output, but also to work within teams to help out where needed, proving useful with necessary input where needed.

Additionally, the team has realised that the project has reached a point where data scraping and model correction are sound and applicable, which is why the push for final results has gained primary importance. This realisation eventually led to a team re-structuring mid-way through the semester into four more focused teams, namely Survey, Register, Demand, and Supply team, in order to focus on applied data analysis and a results-oriented approach rather than on scraping more data and extending existing models. This flexible re-organisation of team structures only proved possible through transparent communication and everyone’s willingness to adapt to evolving project needs. By acknowledging the limitations of previous approaches and embracing a more flexible mindset, the team successfully navigated the transition towards a results-oriented focus, proven by this Progress Report which is structured around a main result that every team has pushed and refined over the course of the last semester.

Last but not least, in terms of academic research and analysis, the learning process was constant and evolving as the different quantitative and qualitative analysis skills of each team member were continuously improved. Team members who initially struggled to translate their findings into an appropriate methodology or into cohesive research findings were encouraged to engage in an open dialogue, thus benefitting from the knowledge sharing of other team members, which helped the team to leverage individual strengths and overcome analytical hurdles.

Therefore, through continuous refinement and a commitment to shared objectives, the team successfully navigated challenges and achieved meaningful progress in research endeavours, proving the team's resilience and commitment to achieving impactful outcomes.

Reference List

Aberdeen City Council. (2016). *HMO Overprovision Consultation Report*. https://committees.aberdeencity.gov.uk/documents/s54919/HMO%20Overprovision%20Policy.pdf

Allinson, J. (2006). Over-educated, over-exuberant and over here? The impact of students on cities. *Planning, Practice&Research*(1), 79–94. https://doi.org/10.1080/02697450600901541

Civic Government. (1982). *Civic Government (Scotland) Act 1982* [Text]. Statute Law Database. https://www.legislation.gov.uk/ukpga/1982/45/contents

CPI Inflation Calculator UK: (2021). *CPI Calculator Data from 1988." CHARTERED SURVEYORS LONDON*. erikasgrig.com/calculators/inflation-calculator-cpi/.

Fife Council. (2022). *Homes in Multiple Occupancy Licence*. https://www.fife.gov.uk/kb/docs/articles/housing/private-rented-sector/homes-in-multiple-occupancy-licence

Fife Council. (2023). *Cabinet Committee—Houses in Multiple Occupations (HMO): Review of the Overprovision Policy* (pp. 25–37). https://www.fife.gov.uk/__data/assets/pdf_file/0037/473779/Agenda-and-Papers-for-Meeting-of-Cabinet-Committee-of-4-May-2023.pdf

Fife Council. (2024). *Fife Council declares housing emergency*. https://www.fife.gov.uk/news/2024/fife-council-declares-housing-emergency

Goodman, A. C., & Thibodeau, T. G. (1998). Housing Market Segmentation. *Journal of Housing Economics*, *7*(2), 121–143. https://doi.org/10.1006/jhec.1998.0229

Gregory, J. J., & Rogerson, J. M. (2019). Housing in multiple occupation and studentification in Johannesburg. *Bulletin of Geography. Socio-Economic Series*, *46*, Article 46. https://doi.org/10.2478/bog-2019-0036

HESA. (2023). *Who's studying in HE? | HESA*. https://www.hesa.ac.uk/data-and-analysis/students/whos-in-he

Hubbard, P. (2008). Regulating the Social Impacts of Studentification: A Loughborough Case Study. *Environment and Planning A: Economy and Space*, *40*(2), 323–341. https://doi.org/10.1068/a396

Munro, M., Turok, I., & Livingston, M. (2009). Students in Cities: A Preliminary Analysis of Their Patterns and Effects. *Environment and Planning A*, *41*, 1805–1825. https://doi.org/10.1068/a41133

Sage, J., Smith, D., & Hubbard, P. (2012). The Diverse Geographies of Studentification: Living Alongside People Not Like Us. *Housing Studies*, *27*(8), 1057–1078. https://doi.org/10.1080/02673037.2012.728570

Saiful Islam, K., & Asami, Y. (2010). Housing market segmentation: A review. *Review of Urban & Regional Development Studies*, *21*, 93–109. https://doi.org/10.1111/j.1467-940X.2009.00161.x

Scottish Parliament. (2006). *Housing (Scotland) Act 2006* [Text]. Statute Law Database. https://www.legislation.gov.uk/asp/2006/1/2022-03-01

Smith, D. (2005). *Studentification: The gentrification fac-tory. In R. Atkinson and G. Bridge (Eds.), Gentrification in a global context: The new urban colonialism (73-90).* London: Routledge.

Smith, D. P., & Holt, L. (2007). Studentification and 'Apprentice' Gentrifiers within Britain's Provincial Towns and Cities: Extending the Meaning of Gentrification. *Environment and Planning A: Economy and Space*, *39*(1), 142–161. https://doi.org/10.1068/a38476

St Andrews QV. (2022). MSP blames council's HMO policy for student accommodation crisis. *St Andrews QV*. https://standrewsqv.org.uk/2022/03/msp-blames-councils-hmo-policy-for-student-accommodation-crisis/

St Andrews QV, S. A. Q. V. (2018). St Andrews HMO problem; where we are and how we got there. *St Andrews QV*. https://standrewsqv.org.uk/2018/11/the-st-andrews-hmo-problem-where-we-are-and-how-we-got-there/

VIP Project. (2023). *2023/24 Martinmas Semester Progress Report 2*.

World Population Review. (2024). *St Andrews Population 2024*. https://worldpopulationreview.com/world-cities/st-andrews-population

Zietz, E., Sirmans, G., MacDonald, L., & Macpherson, D. (2006). The Value of Housing Characteristics: A Meta Analysis. *The Journal of Real Estate Finance and Economics*, *33*, 215–240. https://doi.org/10.1007/s11146-006-9983-5

Appendices

Appendix 1:

```
// Import master data file

import excel "`data'HMO Caps in St Andrews VIP Survey Results", sheet("All Respondents
Cleaned") cellrange(A1:X635) firstrow allstring clear


/////////////////////////////////////////////////////////////////////////////

****General Cleaning****
rename Whatisyourindividualmonthly uncleanrent
rename Howmanyoccupiedbedroomsarei bedrooms
rename Whoisresponsibleforpayingyo responsible
rename Whatisyourfeestatus feestatus
rename whatisyourethnicorigin ethnicorigin
rename whatyeareareyou year
rename whatisyourdegree degree
rename doyouliveinhalls halls
rename doyouliveinauniversitymana unimanaged
rename howdoyoudescribeyourself describeyourself
rename whatisyouraddress address
rename Isyourpropertyrentedthrough propertyrentedthrough
rename ListofCountries countries
rename Byhowmuchwouldyourindividua howmuchrentincreasetomove
rename Haveyoulivedinyourpresenta morethanayear
rename Byhowmuchhasyourrentincrea rentincreased
rename Whatisyourprimarymeansoftr primarytransport
rename Howlongwoulditakeyoutowa howlongtowalk
rename Rankthefollowingfactorswhen ranksafety
rename T rankprice
rename U rankconvenience
rename Imaginea3bedroomfurnishedfl WTPMarketstreet
rename W WTPLamond
rename X WTPDundee

/////////////////////////////////////////////////////////////////////////////

****N/A Cleaning for all variables****
local variablelist1 describeyourself ethnicorigin year degree halls unimanaged address
propertyrentedthrough bedrooms rent feestatus countries responsible
howmuchrentincreasetomove morethanayear rentincreased primarytransport howlongtowalk
ranksafety rankprice rankconvenience WTPMarketstreet WTPLamond WTPDundee
foreach var of local variablelist1 {
    replace `var' = "n/a" if inlist(`var', "N/a", "N/A", "n")
    display "`var'"
}

/////////////////////////////////////////////////////////////////////////////

****Degree Cleaning

*cleans for variables with trailing spaces (one specific chem entry)*

replace degree = lower(degree)

gen subject_a = degree
gen subject_b = degree

local separators "and with & +"
foreach word in `separators' {

    local separatorlength = length("`word'")

    replace subject_a = substr(degree, 1, strpos(degree, " `word' ") - 1) if strpos(
degree, " `word' ") > 0

    replace subject_b = substr(degree, strpos(degree, " `word' ") + `separatorlength' +2,
 .) if strpos(degree, " `word' ") > 0
}
replace subject_b = "" if subject_b == subject_a

*catching common mispellings and variation of subjects (e.g. english literature for
english) regardless of whether it is the degree, first subject or second*
```

```
local subjects "subject_a subject_b degree"
local term_ir "ir"
local term_english "english"
local term_film "film"
local term_finance "finance"
local term_maths "math"
local term_econ "econ"

gen check = 1 if regexm(subject_a, "ir")
foreach subject in subject_a subject_b {
    replace `subject' = "international relations" if strpos(`subject', "`term_ir'") > 0
    replace `subject' = "english" if strpos(`subject', "`term_english'") > 0
    replace `subject' = "mathematics" if strpos(`subject', "`term_maths'") > 0
    replace `subject' = "economics" if strpos(`subject', "`term_econ'") > 0
    replace `subject' = "film" if strpos(`subject', "`term_film'") > 0
    }

    *altering mistaken "33" instead of "3" entry in "what year are you?" question
count if year == "33"
replace year = "3" if year == "33"

///////////////////////////////////////////////////////////////////////////

****Code for finding Walking Distance to Tesco from KP****
save All_Respondents_Analysis_GY_LS_SD.dta, replace
ssc install matchit
ssc install freqindex

import excel "`data'Postcode Continuous Variable 1.xlsx", firstrow clear
gen postcode_merge = substr(Postcode, 6, .)
gen street_merge = substr(Description, 10, .)
save NewPostcode_survey_original.dta, replace
duplicates drop street_merge, force // CAUTION: ONLY FOR USE IF USING STREET FUZZY
MATCHING
drop if street_merge == ""
gen idold = _n
save Postcode_survey.dta, replace

use All_Respondents_Analysis_GY_LS_SD.dta

gen Postcode = " "
local postcodes KY DD ky dd Ky Dd kY dD
foreach postcd in `postcodes' {
    gen `postcd'_check = regexm(address, "`postcd'[0-9]")
    gen `postcd'_position = strpos(address, "`postcd'") if `postcd'_check > 0
    replace Postcode = substr(address, `postcd'_position, 8) if `postcd'_check > 0
    drop `postcd'_position `postcd'_check
}

replace Postcode = "." if Postcode == ""
gen Postcode_upper = upper(Postcode)
drop Postcode
rename Postcode_upper Postcode
gen Postcode_clean = subinstr(Postcode, " ", "", .)
gen Postcode_clean2 = substr(Postcode_clean, 1, 4) + " " + substr(Postcode_clean, 5, .)
drop Postcode
rename Postcode_clean2 Postcode
drop Postcode_clean

gen id_reimport = _n
save Week3_Analysis_KP_PreImport.dta, replace
drop if Postcode != " "
gen idnew = _n
save Week3_Analysis_KP.dta, replace
matchit idnew address using Postcode_survey.dta, idu(idold) txtu(street_merge)
keep if similscore > 0.7

joinby idold using Postcode_survey.dta
joinby idnew using Week3_Analysis_KP.dta

save fuzzy_match.dta, replace

use Week3_Analysis_KP_PreImport.dta

merge 1:m id_reimport using "fuzzy_match.dta", force
gen new_postcode = "KY16 " + postcode_merge
replace Postcode = new_postcode if Postcode == " " & postcode_merge != " "
replace Postcode = "" if Postcode == "KY16 "
drop id_reimport idnew idold similscore Description GridReference Xeasting Ynorthing
Latitude Longitude I J K L M _merge
order address Postcode DistancetoTown street_merge
drop new_postcode
```

```
drop postcode_merge
gen postcode_merge = substr(Postcode, 6, .)
merge m:m postcode_merge using "NewPostcode_survey_original.dta"
drop if _n > 651
drop DistancetoTown _merge
merge m:m Postcode using "NewPostcode_survey_original.dta"
drop check postcode_merge I J K L M _merge
order address Postcode DistancetoTown
drop if _n > 651
*drop if _n > 644
*drop postcode_merge


//////////////////////////////////////////////////////////////////////////////

*Generating Loop for basic descriptive analysis of all variables and encoding into
numerical values
local variablelist2 describeyourself ethnicorigin year degree halls unimanaged address
propertyrentedthrough bedrooms feestatus countries responsible  morethanayear
primarytransport howlongtowalk ranksafety rankprice rankconvenience Postcode subject_a
subject_b
foreach var of local variablelist2 {
    graph pie Freq., over(`var') nodraw
    encode `var', generate (`var'_num)
    label list  `var'_num
}


//// local variablelist3 howmuchrentincreasetomove rent rentincreased WTPMarketstreet
WTPLamond WTPDundee foreach var of local variablelist3 {
////    gen rent_num = .
/// replace rent_num = real(rent) if rent != "n/a"
    ////summarize rent_num}


    gen rent_num = .
    replace rent_num = real(uncleanrent) if uncleanrent != "n/a"
    summarize rent_num

gen rent = .
replace rent = rent_num/9  if rent_num > 3000
replace rent = rent_num if rent_num <3000
label variable rent "Monthly rent"



//////////////////////////////////////////////////////////////////////////////

*Generating Loop to create dummy variables for yes/no questions
local variablelistdummy halls unimanaged morethanayear
foreach var of local variablelistdummy {
    generate dummy_`var' = (`var' == "Yes")

}

//////////////////////////////////////////////////////////////////////////

*Correlations of all Variables
corr describeyourself_num ethnicorigin_num year_num degree_num address_num
propertyrentedthrough_num bedrooms_num feestatus_num countries_num responsible_num
primarytransport_num howlongtowalk_num ranksafety_num rankprice_num rankconvenience_num
Postcode_num dummy_halls dummy_unimanaged dummy_morethanayear subject_a_num subject_b_num

label list Postcode_num
label list feestatus_num
label list bedrooms_num
label list responsible_num
```

```
////////////////////////////////////////////////

*create dummies for bedrooms: n/a is assumed to be studio apartments or halls
generate dummy_bedrooms0 = 1 if (bedrooms == "n/a")
replace dummy_bedrooms0 = 0 if (bedrooms != "n/a")
generate dummy_bedrooms1 = 1 if (bedrooms == "1")
replace dummy_bedrooms1 = 0 if (bedrooms != "1")
generate dummy_bedrooms2 = 1 if (bedrooms == "2")
replace dummy_bedrooms2 = 0 if (bedrooms != "2")
generate dummy_bedrooms3 = 1 if (bedrooms == "3")
replace dummy_bedrooms3 = 0 if (bedrooms != "3")
generate dummy_bedrooms4 = 1 if (bedrooms == "4")
replace dummy_bedrooms4 = 0 if (bedrooms != "4")
generate dummy_bedrooms5 = 1 if (bedrooms == "5")
replace dummy_bedrooms5 = 0 if (bedrooms != "5")
generate dummy_bedrooms6ormore = 1 if (bedrooms == "6 or more")
replace dummy_bedrooms6ormore = 0 if (bedrooms != "6 or more")

////////////////////////////////////////////////

/*
failed to generate
generate dummy_responsibleother = 0 if (responsible !=.)
*/

*create dummy variables for responsible
generate otherfunding = 0
replace otherfunding = 1 if (responsible == "other")

generate n/afunding = 0
replace n/afunding = 1 if (responsible == "n/a")

generate studentloans = 0
replace studentloans = 1 if (responsible == "My student loans go toward paying my rent")

generate ownincome = 0
replace ownincome = 1 if (responsible == "I pay rent with my own income")

generate accomgrant = 0
replace accomgrant = 1 if (responsible == "I have an accommodation grant to pay my rent")

generate familyfunded = 0
replace familyfunded = 1 if (responsible == "A family member or guardian pays my rent")
////////////////////////////////////////////////

*create dummy for property rented through
generate ownproperty = 0
replace ownproperty = 1 if (propertyrentedthrough == "I own the property I live in")
generate lettingagency = 0
replace lettingagency = 1 if (propertyrentedthrough == "Letting Agency")
generate otherpropertytype = 0
replace otherpropertytype = 1 if (propertyrentedthrough == "None of the above. I live in
Halls or..")
generate privatelandlord = 0
replace privatelandlord = 1 if (propertyrentedthrough == "Private Landlord")

////////////////////////////////////////////////

*create dummy for feestatus
generate intlfees = 0
replace intlfees = 1 if (feestatus == "International")
generate scotfees = 0
replace scotfees = 1 if (feestatus == "Scottish")
gen otherfees = 0
replace otherfees = 1 if (feestatus == "other")
generate RUKfees = 0
replace RUKfees = 1 if (feestatus == "rest of the UK")

////////////////////////////////////////////////

gen nonwhite = 0
replace nonwhite = 1 if strpos(ethnicorigin, "White") < 1
*did not create dummy for how much rent increase to move, and ethnic origin

regress DistancetoTown rent
mean rent if rent > 26
tab rent
/////

regress rent RUKfees otherfees intlfees dummy_bedrooms2 dummy_bedrooms3 dummy_bedrooms4
dummy_bedrooms5 dummy_bedrooms6ormore nonwhite dummy_responsibleother
dummy_responsibleownincome dummy_responsiblena dummy_responsiblestudentloans
dummy_responsibleaccomgrant
```

```
*residuals
predict residualprreg_dist, r
histogram residualprreg_dist, normal
kdensity residualprreg_dist, normal

*test skewness
sktest residualprreg_dist

*test heteroskedasticity
estat imtest, white

*test multicolinearity
estat vif

*check functional form
estat ovtest


scatter DistancetoTown rent
corr DistancetoTown rent

scatter rent DistancetoTown
corr rent DistancetoTown

////// Testing impact of first years living in halls

regress rent RUKfees otherfees intlfees dummy_bedrooms2 dummy_bedrooms3 dummy_bedrooms4
dummy_bedrooms5 dummy_bedrooms6ormore nonwhite dummy_responsibleother
dummy_responsibleownincome dummy_responsiblena dummy_responsiblestudentloans
dummy_responsibleaccomgrant if year_num!=1
```

Appendix 2: Variable Cleaning

```
****General Cleaning****
rename Whatisyourindividualmonthly rent
rename Howmanyoccupiedbedroomsarei bedrooms
rename Whoisresponsibleforpayingyo responsible
rename Whatisyourfeestatus feestatus
rename whatisyourethnicorigin ethnicorigin
rename whatyeararewyou year
rename whatisyourdegree degree
rename doyouliveinhalls halls
rename doyouliveinauniversitymana unimanaged
rename howdoyoudescribeyourself describeyourself
rename whatisyouraddress address
rename Isyourpropertyrentedthrough propertyrentedthrough
rename ListofCountries countries
rename Byhowmuchwouldyourindividua howmuchrentincreasetomove
rename Haveyoulivedinyourpresenta morethanayear
rename Byhowmuchhasyourrentincrea rentincreased
rename Whatisyourprimarymeansoftr primarytransport
rename Howlongwouldittakeyoutowa howlongtowalk
rename Rankthefollowingfactorswhen ranksafety
rename T rankprice
rename U rankconvenience
rename Imaginea3bedroomfurnishedfl WTPMarketstreet
rename W WTPLamond
rename X WTPDundee

****N/A Cleaning for all variables****
local variablelist1 describeyourself ethnicorigin year degree halls
unimanaged address propertyrentedthrough bedrooms rent feestatus countries
responsible howmuchrentincreasetomove morethanayear rentincreased
primarytransport howlongtowalk ranksafety rankprice rankconvenience
WTPMarketstreet WTPLamond WTPDundee
foreach var of local variablelist1 {
	replace `var' = "n/a" if inlist(`var', "N/a", "N/A", "n")
	display "`var'"
}
```

```
**altering mistaken "33" instead of "3" entry in "what year are you?"
question
count if year == "33"
replace year = "3" if year == "33"
```

Appendix 3: Finding distance to town using postcode

```
****Code for finding Walking Distance to Tesco from KP****
save All_Respondents_Analysis_GY_LS_SD.dta, replace
ssc install matchit
ssc install freqindex
import excel "`data'Postcode Continuous Variable 1.xlsx", firstrow clear
gen postcode_merge = substr(Postcode, 6, .)
gen street_merge = substr(Description, 10, .)
save NewPostcode_survey_original.dta, replace
duplicates drop street_merge, force // CAUTION: ONLY FOR USE IF USING
STREET FUZZY MATCHING
drop if street_merge == ""
gen idold = _n
save Postcode_survey.dta, replace
use All_Respondents_Analysis_GY_LS_SD.dta
gen Postcode = " "
local postcodes KY DD ky dd Ky Dd kY dDforeach postcd in `postcodes' {
gen `postcd'_check = regexm(address, "`postcd'[0-9]")
gen `postcd'_position = strpos(address, "`postcd'") if `postcd'_check > 0
replace Postcode = substr(address, `postcd'_position, 8) if `postcd'_check
> 0
drop `postcd'_position `postcd'_check
}
replace Postcode = "." if Postcode == ""
gen Postcode_upper = upper(Postcode)
drop Postcode
rename Postcode_upper Postcode
gen Postcode_clean = subinstr(Postcode, " ", "", .)
gen Postcode_clean2 = substr(Postcode_clean, 1, 4) + " " +
substr(Postcode_clean, 5, .)
drop Postcode
rename Postcode_clean2 Postcode
drop Postcode_clean

gen id_reimport = _n
save Week3_Analysis_KP_PreImport.dta, replace
drop if Postcode != " "
gen idnew = _n
save Week3_Analysis_KP.dta, replace
matchit idnew address using Postcode_survey.dta, idu(idold)
txtu(street_merge)
keep if similscore > 0.7

joinby idold using Postcode_survey.dta
joinby idnew using Week3_Analysis_KP.dta

save fuzzy_match.dta, replace

use Week3_Analysis_KP_PreImport.dta

merge 1:m id_reimport using "fuzzy_match.dta", force
gen new_postcode = "KY16 " + postcode_merge
replace Postcode = new_postcode if Postcode == " " & postcode_merge != " "
replace Postcode = "" if Postcode == "KY16 "
```

```
drop id_reimport idnew idold similscore Description GridReference Xeasting
Ynorthing Latitude Longitude I J K L M _merge
order address Postcode DistancetoTown street_merge
drop new_postcode

drop postcode_merge
gen postcode_merge = substr(Postcode, 6, .)
merge m:m postcode_merge using "NewPostcode_survey_original.dta"
drop if _n > 651
drop DistancetoTown _merge
merge m:m Postcode using "NewPostcode_survey_original.dta"
drop check postcode_merge I J K L M _merge
order address Postcode DistancetoTown
drop if _n > 651
*drop if _n > 644
*drop postcode_merge
```

Appendix 4: Transforming variables

```
*Generating Loop to encode variables into categories
local variablelist2 describeyourself ethnicorigin year degree halls
unimanaged address propertyrentedthrough bedrooms feestatus countries
responsible  morethanayear  primarytransport howlongtowalk ranksafety
rankprice rankconvenience Postcode subject_a subject_b
foreach var of local variablelist2 {
graph pie Freq., over(`var') nodraw
encode `var', generate (`var'_num)
label list  `var'_num
}

*Transforming halls rent to monthly values
gen rent_month = .
replace rent_month = rent_num/9  if rent_num > 3000
replace rent_month = rent_num if rent_num <3000
label variable rent_month "Monthly rent"
*Generating Loop to create dummy variables for yes/no questions
local variablelistdummy halls unimanaged morethanayear
foreach var of local variablelistdummy {
generate dummy_`var' = (`var' == "Yes")
}

////////////////////////////////////////////////

*create dummies for bedrooms: n/a is assumed to be studio apartments or
halls
generate dummy_bedrooms0 = 1 if (bedrooms == "n/a")
replace dummy_bedrooms0 = 0 if (bedrooms != "n/a")
generate dummy_bedrooms1 = 1 if (bedrooms == "1")
replace dummy_bedrooms1 = 0 if (bedrooms != "1")
generate dummy_bedrooms2 = 1 if (bedrooms == "2")
replace dummy_bedrooms2 = 0 if (bedrooms != "2")
generate dummy_bedrooms3 = 1 if (bedrooms == "3")
replace dummy_bedrooms3 = 0 if (bedrooms != "3")
generate dummy_bedrooms4 = 1 if (bedrooms == "4")
replace dummy_bedrooms4 = 0 if (bedrooms != "4")
generate dummy_bedrooms5 = 1 if (bedrooms == "5")
replace dummy_bedrooms5 = 0 if (bedrooms != "5")
generate dummy_bedrooms6ormore = 1 if (bedrooms == "6 or more")
replace dummy_bedrooms6ormore = 0 if (bedrooms != "6 or more")
```

```
/////////////////////////////////////////////////

/*
failed to generate
generate dummy_responsibleother = 0 if (responsible !=.)
*/

*create dummy variables for responsible
generate dummy_responsibleother = 0
replace dummy_responsibleother = 1 if (responsible == "other")
generate dummy_responsiblena = 0
replace dummy_responsiblena = 1 if (responsible == "n/a")
generate dummy_responsiblestudentloans = 0
replace dummy_responsiblestudentloans = 1 if (responsible == "My student
loans go toward paying my rent")
generate dummy_responsibleownincome = 0
replace dummy_responsibleownincome = 1 if (responsible == "I pay rent with
my own income")
generate dummy_responsibleaccomgrant = 0
replace dummy_responsibleaccomgrant = 1 if (responsible == "I have an
accommodation grant to pay my rent")
generate dummy_responsiblefamily = 0
replace dummy_responsiblefamily = 1 if (responsible == "A family member or
guardian pays my rent")

/////////////////////////////////////////////////

*create dummy for property rented through
generate dummy_propertyrentedowned = 0
replace dummy_propertyrentedowned = 1 if (propertyrentedthrough == "I own
the property I live in")
generate dummy_propertyrentedagency = 0
replace dummy_propertyrentedagency = 1 if (propertyrentedthrough ==
"Letting Agency")
generate dummy_propertyrentednone = 0
replace dummy_propertyrentednone = 1 if (propertyrentedthrough == "None of
the above. I live in Halls or..")
generate dummy_propertyrentedprivate = 0
replace dummy_propertyrentedprivate = 1 if (propertyrentedthrough ==
"Private Landlord")

/////////////////////////////////////////////////

*create dummy for feestatus
generate dummy_feestatusinternational = 0
replace dummy_feestatusinternational = 1 if (feestatus == "International")
generate dummy_feestatusscottish = 0
replace dummy_feestatusscottish = 1 if (feestatus == "Scottish")
gen dummy_feestatusother = 0
replace dummy_feestatusother = 1 if (feestatus == "other")
generate dummy_feestatusrestUK = 0
replace dummy_feestatusrestUK = 1 if (feestatus == "rest of the UK")

/////////////////////////////////////////////////

gen dummy_nonwhite = 0
replace dummy_nonwhite = 1 if strpos(ethnicorigin, "White") < 1
```

Appendix 5: Data Cleaning Codes for the Official Council Data (Produces FOIRegister_Cleaned.dta)

```
// Importing distance to town information
import excel "`data'Postcode Continuous Variable 1.xlsx", firstrow clear
gen postcode_merge = substr(Postcode, 1, .)
save Postcode_survey.dta, replace

// Set working directory

cd "`data'"

// Import dataset

import excel "`data'All HMO Apps.xlsx", firstrow clear

// Basic data cleaning to obtain result for PR1

* Removing addresses outside St Andrews
tab Ward
drop if Ward != "St Andrews"

* Extracting postcode from Address
gen Postcode = " "
gen KY16_check = regexm(HMOAddress, "KY16")
gen KY16_position = strpos(HMOAddress, "KY16") if KY16_check > 0
replace Postcode = upper(substr(HMOAddress, KY16_position, 8)) if KY16_check > 0
drop KY16_check KY16_position
order HMOAddress Postcode

count if Postcode == " " // Missing 1,115 Postcode observations

* Assigning missing postcodes systematically
replace Postcode = "KY16 9LY" if strpos(HMOAddress, "David Russel") > 0 & Postcode == " " // Assigning a postcode
to David Russell Apts
count if Postcode == " " // Missing Postcode observations reduced to 142
replace Postcode = "KY16 9UE" if strpos(HMOAddress, "Fife Park") > 0 & Postcode == " " // Assigning a postcode to
Fife Park
count if Postcode == " " // Missing Postcode observations reduced to 38
replace Postcode = "KY16 9LY" if strpos(HMOAddress, "Buchanan Gardens") > 0 & Postcode == " " // Assigning a
postcode to Buchanan Gardens
count if Postcode == " " // Missing Postcode observations reduced to 17
replace Postcode = "KY16 9QW" if strpos(HMOAddress, "South Street") > 0 & Postcode == " " // Assigning a postcode
to South Street
count if Postcode == " " // Missing Postcode observations reduced to 12
replace Postcode = "KY16 9NS" if strpos(HMOAddress, "Market Street") > 0 & Postcode == " " // Assigning a
postcode to Market Street
count if Postcode == " " // Missing Postcode observations reduced to 9
* Assigning missing postcodes manually
replace Postcode = "KY16 8RQ" if AppRef == "F1178/9" & Postcode == " "
replace Postcode = "KY16 9DL" if AppRef == "113" & Postcode == " "
replace Postcode = "KY16 9DY" if AppRef == "F1667/10" & Postcode == " "
replace Postcode = "KY16 9FB" if AppRef == "906" & Postcode == " "
replace Postcode = "KY16 9JA" if AppRef == "105" & Postcode == " "
replace Postcode = "KY16 8AA" if AppRef == "F1586/10" & Postcode == " "
replace Postcode = "KY16 8AD" if AppRef == "F1136/8" & Postcode == " "
replace Postcode = "KY16 9DT" if AppRef == "246" & Postcode == " "
replace Postcode = "KY16 9DG" if AppRef == "310" & Postcode == " "
count if Postcode == " " // No missing postcodes

* Removing rejected/withdrawn/pending applications
tab LISTAT
tab StatusDesc
drop if StatusDesc == "Withdrawn"
drop if StatusDesc == "Refused"
drop if StatusDesc == "Checks Outstanding"
drop if StatusDesc == "Consultees Outstanding"
drop if LISTAT == "4_DEC"

// Importing DistancetoTown information

gen postcode_merge = substr(Postcode, 1, .)
merge m:1 postcode_merge using "Postcode_survey.dta"
drop if _merge == 2 // Drop any imported observations that have not been matched
drop I J K L M _merge
order Postcode DistancetoTown
count if DistancetoTown == . // 17 missing observations

*Assigning DistancetoTown systematically
drop if Postcode == "KY16 9SQ" // located outside St Andrews (Kincaple)
replace DistancetoTown = 0.4 if Postcode == "KY16 9NZ" // non-existent postcode - supposed to be KY16 9HZ
replace Postcode = "KY16 9HZ" if Postcode == "KY16 9NZ" // as above
drop if Postcode == "KY16 8NT" // located outside St Andrews (Mount Melville)

*Assigning DistancetoTown manually
replace DistancetoTown = 0.35 if AppRef == "F1820/11" & DistancetoTown == . // Postcode does not show up -
perhaps a historically recognised postcode?
replace Postcode = "KY16 9PB" if AppRef == "F1198/9" // Postcode misformatted (contains !)
replace DistancetoTown = 0.1 if AppRef == "F1198/9"
drop if Postcode == "KY16 8LJ" // Outside St Andrews
drop if Postcode == "KY16 8LP" // Outside St Andrews
count if DistancetoTown == . // 0 missing observations
```

```
// Converting to long form (based on Week4_ExpeditedPDFData_FW)

save FOIData, replace
clear
set obs 1
gen month = 1
expand 12 * (2026 - 2002 + 1)
bysort month: gen year = 2002 + floor((_n-1)/12)
replace month = mod(_n-1, 12) + 1
gen ym = ym(year, month)
gen ym_date = dofm(ym)
tempfile monthlist
save "`monthlist'"
use "FOIData.dta"
tempfile licenses
save "`licenses'"
cross using "`monthlist'"
gen Active = (Issued <= ym_date) & (ym_date <= Expiry)

// Saving the cleaned FOI Register
save "`data'FOIRegister_Cleaned", replace
```

Appendix 6: Data Cleaning Codes for the Official Council Data (Produces FOIRegister_UniRemoved_Cleaned.dta)

```
// Importing distance to town information
import excel "`data'Postcode Continuous Variable 1.xlsx", firstrow clear
gen postcode_merge = substr(Postcode, 1, .)
save Postcode_survey.dta, replace

// Set working directory

cd "`data'"

// Import dataset

import excel "`data'All HMO Apps.xlsx", firstrow clear

// Basic data cleaning to obtain result for PR1

* Removing addresses outside St Andrews
tab Ward
drop if Ward != "St Andrews"

* Extracting postcode from Address
gen Postcode = " "
gen KY16_check = regexm(HMOAddress, "KY16")
gen KY16_position = strpos(HMOAddress, "KY16") if KY16_check > 0
replace Postcode = upper(substr(HMOAddress, KY16_position, 8)) if KY16_check > 0
drop KY16_check KY16_position
order HMOAddress Postcode

count if Postcode == " " // Missing 1,115 Postcode observations

* Assigning missing postcodes systematically
replace Postcode = "KY16 9LY" if strpos(HMOAddress, "David Russel") > 0 & Postcode == " " // Assigning a postcode
to David Russell Apts
count if Postcode == " " // Missing Postcode observations reduced to 142
replace Postcode = "KY16 9UE" if strpos(HMOAddress, "Fife Park") > 0 & Postcode == " " // Assigning a postcode to
Fife Park
count if Postcode == " " // Missing Postcode observations reduced to 38
replace Postcode = "KY16 9LY" if strpos(HMOAddress, "Buchanan Gardens") > 0 & Postcode == " " // Assigning a
postcode to Buchanan Gardens
count if Postcode == " " // Missing Postcode observations reduced to 17
replace Postcode = "KY16 9QW" if strpos(HMOAddress, "South Street") > 0 & Postcode == " " // Assigning a postcode
to South Street
count if Postcode == " " // Missing Postcode observations reduced to 12
replace Postcode = "KY16 9NS" if strpos(HMOAddress, "Market Street") > 0 & Postcode == " " // Assigning a
postcode to Market Street
count if Postcode == " " // Missing Postcode observations reduced to 9

* Assigning missing postcodes manually
replace Postcode = "KY16 8RQ" if AppRef == "F1178/9" & Postcode == " "
replace Postcode = "KY16 9DL" if AppRef == "113" & Postcode == " "
replace Postcode = "KY16 9DY" if AppRef == "F1667/10" & Postcode == " "
replace Postcode = "KY16 9FB" if AppRef == "906" & Postcode == " "
replace Postcode = "KY16 9JA" if AppRef == "105" & Postcode == " "
replace Postcode = "KY16 8AA" if AppRef == "F1586/10" & Postcode == " "
replace Postcode = "KY16 8AD" if AppRef == "F1136/8" & Postcode == " "
replace Postcode = "KY16 9DT" if AppRef == "246" & Postcode == " "
replace Postcode = "KY16 9DG" if AppRef == "310" & Postcode == " "
count if Postcode == " " // No missing postcodes

* Removing rejected/withdrawn/pending applications
tab LISTAT
tab StatusDesc
drop if StatusDesc == "Withdrawn"
drop if StatusDesc == "Refused"
drop if StatusDesc == "Checks Outstanding"
drop if StatusDesc == "Consultees Outstanding"
drop if LISTAT == "4_DEC"

// Importing DistancetoTown information

gen postcode_merge = substr(Postcode, 1, .)
merge m:1 postcode_merge using "Postcode_survey.dta"
drop if _merge == 2 // Drop any imported observations that have not been matched
drop I J K L M _merge
order Postcode DistancetoTown
count if DistancetoTown == . // 17 missing observations

*Assigning DistancetoTown systematically
drop if Postcode == "KY16 9SQ" // located outside St Andrews (Kincaple)
replace DistancetoTown = 0.4 if Postcode == "KY16 9NZ" // non-existent postcode - supposed to be KY16 9HZ
replace Postcode = "KY16 9HZ" if Postcode == "KY16 9NZ" // as above
drop if Postcode == "KY16 8NT" // located outside St Andrews (Mount Melville)

*Assigning DistancetoTown manually
replace DistancetoTown = 0.35 if AppRef == "F1820/11" & DistancetoTown == . // Postcode does not show up -
perhaps a historically recognised postcode?
replace Postcode = "KY16 9PB" if AppRef == "F1198/9" // Postcode misformatted (contains !)
replace DistancetoTown = 0.1 if AppRef == "F1198/9"
drop if Postcode == "KY16 8LJ" // Outside St Andrews
drop if Postcode == "KY16 8LP" // Outside St Andrews
count if DistancetoTown == . // 0 missing observations
```

```
// Removing University halls and Ayton House
drop if strpos(HMOAddress, "David Russell") > 0
drop if strpos(HMOAddress, "Albany Park") > 0
drop if strpos(HMOAddress, "Gannochy House") > 0
drop if strpos(HMOAddress, "New Hall") > 0
drop if strpos(HMOAddress, "Agnes Blackadder") > 0
drop if strpos(HMOAddress, "Andrew Melville Hall") > 0
drop if strpos(HMOAddress, "Mcintosh Hall") > 0
drop if strpos(HMOAddress, "McIntosh Hall") > 0
drop if strpos(HMOAddress, "Powell Hall") > 0
drop if strpos(HMOAddress, "St Salvators Hall") > 0
drop if strpos(HMOAddress, "St Salvator's Hall") > 0
drop if strpos(HMOAddress, "Whitehorn Hall") > 0
drop if strpos(HMOAddress, "St Salvators Chapel") > 0 // Also removing chapel
drop if strpos(HMOAddress, "University Hall") > 0
drop if strpos(HMOAddress, "St Regulus Hall") > 0
drop if strpos(HMOAddress, "St Regulus Annexe") > 0
drop if strpos(HMOAddress, "Hamilton Hall") > 0
drop if strpos(HMOAddress, "John Burnet Hall") > 0
drop if strpos(HMOAddress, "John Burnett Hall") > 0
drop if strpos(HMOAddress, "Fife Park") > 0
drop if strpos(HMOAddress, "House") > 0 & strpos(Description, "KY16 9LY Buchanan Gardens") > 0
sort TotOccs

// Converting to long form (based on Week4_ExpeditedPDFData_FW)

save FOIData, replace
clear
set obs 1
gen month = 1
expand 12 * (2026 - 2002 + 1)
bysort month: gen year = 2002 + floor((_n-1)/12)
replace month = mod(_n-1, 12) + 1
gen ym = ym(year, month)
gen ym_date = dofm(ym)
tempfile monthlist
save "`monthlist'"
use "FOIData.dta"
tempfile licenses
save "`licenses'"
cross using "`monthlist'"
gen Active = (Issued <= ym_date) & (ym_date <= Expiry)

// Saving the cleaned FOI Register
save "`data'FOIRegister_UniRemoved_Cleaned", replace
```

Appendix 7: Property Level Conversion Code

```
// Creating the matching variable 'address_extracted'
gen address_extracted = substr(HMOAddress, 1, strpos(HMOAddress, "St Andrews") -1)
replace address_extracted = substr(HMOAddress, 1, strpos(HMOAddress, "KY") -1) if missing(address_extracted)
replace address_extracted = subinstr(address_extracted, " ", "", .)


replace address_extracted = substr(address_extracted, strpos(address_extracted, "Flat") - 2,.) if strpos(
address_extracted, "Flat") > 3
replace address_extracted = substr(address_extracted, strpos(address_extracted, "Flat"),.) if strpos(
address_extracted, "Flat") > 0 & strpos(address_extracted, "Flat") < 4
replace address_extracted = "A" + address_extracted if strpos(HMOAddress, "Attic") > 0
replace address_extracted = "F" + address_extracted if strpos(HMOAddress, "First") > 0
replace address_extracted = "M" + address_extracted if strpos(HMOAddress, "Middle") > 0
replace address_extracted = "U" + address_extracted if strpos(HMOAddress, "Upper") > 0

replace address_extracted = upper(address_extracted)
replace address_extracted = subinstr(address_extracted, "`=char(13)'", "", .) // Remove line breaks
replace address_extracted = subinstr(address_extracted, "`=char(10)'", "", .) // Remove line breaks
sort address_extracted
order HMOAddress address_extracted id

// Removing some additional problematic properties (most university halls)
drop if strpos(address_extracted, "DEANSCOURT") > 0
drop if strpos(address_extracted, "DEANCOURT") > 0

// Matching properties
gen match_ident = 0

** The automatic matching is not flawless - we can manually match some addresses before running the matching
algorithm by adding the obs number of the obs_list local variable. This is generally only necessary for
properties with text before the house number
local id_list 3565
foreach identifier of local id_list {
replace address_extracted = address_extracted[_n-1] if id==`identifier'
}

** Our matching criteria is the first 12 letters of the address_extracted var matching, or the first six
letters matching if there are fewer than 12 characters
replace match_ident = match_ident[_n-1]+1 if substr(address_extracted, 1, 12) == substr(address_extracted[_n-
1], 1, 12)
replace match_ident = match_ident[_n-1]+1 if substr(address_extracted, 1, 6) == substr(address_extracted[_n-1
], 1, 6) & strlen(address_extracted) < 12
summarize match_id
```

```
////////////////////////////////////////////////////////////////

** This should be our approach to manually checking for errors (false positives)
gen HMOAddressnospace = subinstr(HMOAddress, " ", "", .)
gen check = 1 if match_ident != 0 & HMOAddressnospace != HMOAddressnospace[_n-1]
order check match_ident
tab check // 873 possible mistakes to check (very high false positive)

///////////////////////////////////////////////////////////////

// Converting from license-level to property-level
drop address_extracted HMOAddressnospace
order match_id HMOAddress Postcode DistancetoTown Ward TotOccs Description GridReference Xeasting Ynorthing
Latitude Longitude

// Renaming the original license variables and generating new variables
foreach var in AppRef LISTAT StatusDesc Decision {
    rename `var' `var'_0
    forval i = 1/9 {
        gen `var'_`i' = ""
    }
}

foreach var in Received Issued Expiry {
    rename `var' `var'_0
    forval i = 1/9 {
        gen `var'_`i' = .
        format `var'_`i' %td
    }
}

foreach var in AppRef LISTAT StatusDesc Received Issued Expiry Decision {
forval i = 1/9 {
    replace `var'_`i' = `var'_0 if match_id == `i'
    replace `var'_`i' = `var'_`i'[_n+`i'-1] if match_id[_n+`i'-1] != 0
    replace `var'_`i' = `var'_`i'[_n+1] if match_id == 0 & match_id[_n+1] != 0
}
}

// Tidying and reordering
drop if match_id != 0
drop match_id
sort Postcode
```

```
////////////////////////////////////////////////////////////////

gen id_failedmatchtest = _n

** This should be our approach to manually checking for errors (failed matches)
order id_failedmatchtest HMOAddress // Manually check for failed matches within same postcode_merge

replace id_failedmatchtest = 17 if id_failedmatchtest == 1
replace id_failedmatchtest = 18 if id_failedmatchtest == 5
replace id_failedmatchtest = 9 if id_failedmatchtest == 8
replace id_failedmatchtest = 21 if id_failedmatchtest == 10
replace id_failedmatchtest = 27 if id_failedmatchtest == 25
replace id_failedmatchtest = 29 if id_failedmatchtest == 26
replace id_failedmatchtest = 31 if id_failedmatchtest == 28
replace id_failedmatchtest = 49 if id_failedmatchtest == 47
replace id_failedmatchtest = 89 if id_failedmatchtest == 85
replace id_failedmatchtest = 87 if id_failedmatchtest == 86
replace id_failedmatchtest = 105 if id_failedmatchtest == 98
replace id_failedmatchtest = 237 if id_failedmatchtest == 235
replace id_failedmatchtest = 238 if id_failedmatchtest == 236
replace id_failedmatchtest = 242 if id_failedmatchtest == 239
replace id_failedmatchtest = 282 if id_failedmatchtest == 281
replace id_failedmatchtest = 290 if id_failedmatchtest == 289
replace id_failedmatchtest = 337 if id_failedmatchtest == 336
replace id_failedmatchtest = 341 if id_failedmatchtest == 340
replace id_failedmatchtest = 398 if id_failedmatchtest == 394
replace id_failedmatchtest = 399 if id_failedmatchtest == 396
replace id_failedmatchtest = 406 if id_failedmatchtest == 405
replace id_failedmatchtest = 433 if id_failedmatchtest == 439
replace id_failedmatchtest = 440 if id_failedmatchtest == 438
replace id_failedmatchtest = 450 if id_failedmatchtest == 449
replace id_failedmatchtest = 457 if id_failedmatchtest == 453
replace id_failedmatchtest = 459 if id_failedmatchtest == 455
replace id_failedmatchtest = 460 if id_failedmatchtest == 461
replace id_failedmatchtest = 490 if id_failedmatchtest == 495
replace id_failedmatchtest = 492 if id_failedmatchtest == 498
replace id_failedmatchtest = 502 if id_failedmatchtest == 503
replace id_failedmatchtest = 511 if id_failedmatchtest == 514
replace id_failedmatchtest = 512 if id_failedmatchtest == 519
replace id_failedmatchtest = 515 if id_failedmatchtest == 517
replace id_failedmatchtest = 540 if id_failedmatchtest == 544
replace id_failedmatchtest = 549 if id_failedmatchtest == 550
replace id_failedmatchtest = 557 if id_failedmatchtest == 564
replace id_failedmatchtest = 584 if id_failedmatchtest == 589
replace id_failedmatchtest = 593 if id_failedmatchtest == 594
replace id_failedmatchtest = 596 if id_failedmatchtest == 597
replace id_failedmatchtest = 615 if id_failedmatchtest == 619
replace id_failedmatchtest = 617 if id_failedmatchtest == 620
replace id_failedmatchtest = 624 if id_failedmatchtest == 627
replace id_failedmatchtest = 533 if id_failedmatchtest == 635
replace id_failedmatchtest = 641 if id_failedmatchtest == 650
replace id_failedmatchtest = 647 if id_failedmatchtest == 645
replace id_failedmatchtest = 685 if id_failedmatchtest == 686
replace id_failedmatchtest = 693 if id_failedmatchtest == 698
replace id_failedmatchtest = 701 if id_failedmatchtest == 707
replace id_failedmatchtest = 706 if id_failedmatchtest == 709
replace id_failedmatchtest = 717 if id_failedmatchtest == 718
replace id_failedmatchtest = 721 if id_failedmatchtest == 724
```

```
replace id_failedmatchtest = 723 if id_failedmatchtest == 729
replace id_failedmatchtest = 730 if id_failedmatchtest == 734
replace id_failedmatchtest = 739 if id_failedmatchtest == 740
replace id_failedmatchtest = 743 if id_failedmatchtest == 744
replace id_failedmatchtest = 776 if id_failedmatchtest == 777
replace id_failedmatchtest = 795 if id_failedmatchtest == 796
replace id_failedmatchtest = 806 if id_failedmatchtest == 807
replace id_failedmatchtest = 817 if id_failedmatchtest == 818
replace id_failedmatchtest = 825 if id_failedmatchtest == 831
replace id_failedmatchtest = 829 if id_failedmatchtest == 830
replace id_failedmatchtest = 832 if id_failedmatchtest == 837
replace id_failedmatchtest = 848 if id_failedmatchtest == 850
replace id_failedmatchtest = 857 if id_failedmatchtest == 858
replace id_failedmatchtest = 867 if id_failedmatchtest == 869
replace id_failedmatchtest = 876 if id_failedmatchtest == 879
replace id_failedmatchtest = 881 if id_failedmatchtest == 890
replace id_failedmatchtest = 927 if id_failedmatchtest == 930
replace id_failedmatchtest = 947 if id_failedmatchtest == 948
replace id_failedmatchtest = 966 if id_failedmatchtest == 974
replace id_failedmatchtest = 968 if id_failedmatchtest == 977
replace id_failedmatchtest = 979 if id_failedmatchtest == 980
replace id_failedmatchtest = 986 if id_failedmatchtest == 987

sort id_failedmatchtest

foreach var in AppRef LISTAT StatusDesc Decision {
    forval i = 0/7 {
        gen `var'_m`i' = ""
    replace `var'_m`i' = `var'_`i'[_n+1] if id_failedmatchtest == id_failedmatchtest[_n+1]
}
}

foreach var in Received Issued Expiry {
    forval i = 0/7 {
        gen `var'_m`i' = .
        format `var'_m`i' %td
    replace `var'_m`i' = `var'_`i'[_n+1] if id_failedmatchtest == id_failedmatchtest[_n+1]
}
}
```

```
foreach var in AppRef LISTAT StatusDesc Decision {
    forval i = 0/7 {
        replace `var'_1 = `var'_m`i' if `var'_1 == ""
        replace `var'_2 = `var'_m`i' if `var'_2 == "" & `var'_1 != `var'_m`i'
        replace `var'_3 = `var'_m`i' if `var'_3 == "" & `var'_1 != `var'_m`i' & `var'_2 != `var'_m`i' & `var'
_2 != `var'_m`i'
        replace `var'_4 = `var'_m`i' if `var'_4 == "" & `var'_1 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i'
        replace `var'_5 = `var'_m`i' if `var'_5 == "" & `var'_1 != `var'_m`i' & `var'_4 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'_4 != `var'_m`i'
        replace `var'_6 = `var'_m`i' if `var'_6 == "" & `var'_1 != `var'_m`i' & `var'_5 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'_4 != `var'_m`i' & `var'_5 != `var'_m`i'
        replace `var'_7 = `var'_m`i' if `var'_7 == "" & `var'_1 != `var'_m`i' & `var'_6 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'_4 != `var'_m`i' & `var'_5 != `var'_m`i' & `var'_6 != `var'_m
`i'
}
}

foreach var in Received Issued Expiry {
    forval i = 0/7 {
        replace `var'_1 = `var'_m`i' if `var'_1 == .
        replace `var'_2 = `var'_m`i' if `var'_2 == . & `var'_1 != `var'_m`i'
        replace `var'_3 = `var'_m`i' if `var'_3 == . & `var'_1 != `var'_m`i' & `var'_2 != `var'_m`i' & `var'
_2 != `var'_m`i'
        replace `var'_4 = `var'_m`i' if `var'_4 == . & `var'_1 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i'
        replace `var'_5 = `var'_m`i' if `var'_5 == . & `var'_1 != `var'_m`i' & `var'_4 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'_4 != `var'_m`i'
        replace `var'_6 = `var'_m`i' if `var'_6 == . & `var'_1 != `var'_m`i' & `var'_5 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'_4 != `var'_m`i' & `var'_5 != `var'_m`i'
        replace `var'_7 = `var'_m`i' if `var'_7 == . & `var'_1 != `var'_m`i' & `var'_6 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'_4 != `var'_m`i' & `var'_5 != `var'_m`i' & `var'_6 != `var'_m
`i'
}
}

foreach var in AppRef LISTAT StatusDesc Decision Received Issued Expiry {
    drop `var'_8 `var'_9
    forval i = 0/7 {
        drop `var'_m`i'
    }
}
```

```
// Final manual matching of properties with more than a single duplicate

replace id_failedmatchtest = 540 if id_failedmatchtest == 541
replace id_failedmatchtest = 557 if id_failedmatchtest == 569
replace id_failedmatchtest = 817 if id_failedmatchtest == 819

sort id_failedmatchtest

foreach var in AppRef LISTAT StatusDesc Decision {
    forval i = 0/7 {
        gen `var'_m`i' = ""
    replace `var'_m`i' = `var'_`i'[_n+1] if id_failedmatchtest == id_failedmatchtest[_n+1]
}
}

foreach var in Received Issued Expiry {
    forval i = 0/7 {
        gen `var'_m`i' = .
        format `var'_m`i' %td
    replace `var'_m`i' = `var'_`i'[_n+1] if id_failedmatchtest == id_failedmatchtest[_n+1]
}
}

foreach var in AppRef LISTAT StatusDesc Decision {
    forval i = 0/7 {
        replace `var'_1 = `var'_m`i' if `var'_1 == ""
        replace `var'_2 = `var'_m`i' if `var'_2 == "" & `var'_1 != `var'_m`i'
        replace `var'_3 = `var'_m`i' if `var'_3 == "" & `var'_1 != `var'_m`i' & `var'_2 != `var'_m`i' & `var'
_2 != `var'_m`i'
        replace `var'_4 = `var'_m`i' if `var'_4 == "" & `var'_1 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i'
        replace `var'_5 = `var'_m`i' if `var'_5 == "" & `var'_1 != `var'_m`i' & `var'_4 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'_4 != `var'_m`i'
        replace `var'_6 = `var'_m`i' if `var'_6 == "" & `var'_1 != `var'_m`i' & `var'_5 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'_4 != `var'_m`i' & `var'_5 != `var'_m`i'
        replace `var'_7 = `var'_m`i' if `var'_7 == "" & `var'_1 != `var'_m`i' & `var'_6 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'_4 != `var'_m`i' & `var'_5 != `var'_m`i' & `var'_6 != `var'_m
`i'
}
}

foreach var in Received Issued Expiry {
    forval i = 0/7 {
        replace `var'_1 = `var'_m`i' if `var'_1 == .
        replace `var'_2 = `var'_m`i' if `var'_2 == . & `var'_1 != `var'_m`i'
        replace `var'_3 = `var'_m`i' if `var'_3 == . & `var'_1 != `var'_m`i' & `var'_2 != `var'_m`i' & `var'
_2 != `var'_m`i'
        replace `var'_4 = `var'_m`i' if `var'_4 == . & `var'_1 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i'
        replace `var'_5 = `var'_m`i' if `var'_5 == . & `var'_1 != `var'_m`i' & `var'_4 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'_4 != `var'_m`i'
        replace `var'_6 = `var'_m`i' if `var'_6 == . & `var'_1 != `var'_m`i' & `var'_5 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'_4 != `var'_m`i' & `var'_5 != `var'_m`i'
        replace `var'_7 = `var'_m`i' if `var'_7 == . & `var'_1 != `var'_m`i' & `var'_6 != `var'_m`i' & `var'
_2 != `var'_m`i' & `var'_3 != `var'_m`i' & `var'_4 != `var'_m`i' & `var'_5 != `var'_m`i' & `var'_6 != `var'_m
`i'
}
}

foreach var in AppRef LISTAT StatusDesc Decision Received Issued Expiry {
    forval i = 0/7 {
        drop `var'_m`i'
    }
}

drop if id_failedmatchtest[_n-1] == id_failedmatchtest[_n]

save "`data'FOI_Register_PropLevel_Final.dta"
```

Appendix 8: PDF Conversion Codes producing "Expedited_2012Onwards.dta"

```
/*---------------------------------
|                                 |
|    Section 1: Identification    |
|                                 |
|--------------------------------*/

* creates serial number
g sr_no = _n

* creates pattern breaker for future id of cells that break patterns
g problem = ""


* creates id variable in sumedh like style. Assigns some basic info to id
g id = ""

* establishes if a cell contains contact details
replace id = "Contact" if var1 == "HMO Public Register" | var1 == "HMOLicensingContactDetails" | var1 ==
"HMOLicensing,PrivateSectorHousingTeam," | var1 == "Tel:HMOLicensingon01592583162or" | var1 ==
"Email:HMO.licensing@fife.gov.uk" | var1 == "3rdFloor,RothesayHouse,RothesayPlace," | var1 ==
"Glenrothes,Fife,KY75PQ" | var1 == "HMOLicensing,PrivateSectorHousingTeam, HMO Public Register
3rdFloor,RothesayHouse,RothesayPlace," | var1 == "App Ref Number Applicant Name/s WARD HMO Address Agent Name
License Status Date of Ap Date Issued Expire Date Tot Occs Decision"
replace id = "Contact" in 1

forval x = 2012/2023{
    replace id = "Contact" if var1 == "`x'"
}

* indicates if a cell is empty --> use for detecting page breaks
replace id = "Empty" if var1 == ""

* Indicates if its the first row and extracts the licence information and assigns it to variable
replace id = "First Row" if regexm(var1, "F0[^ ]+")
replace id = "First Row" if regexm(var1, "F2[^ ]+")

* Code to create pdf page number this is for the old pdf and so it starts of pg 577 since the preceeding pages
are of interest
gen pdfPageNum = 0
replace pdfPageNum = 577 in 1
replace pdfPageNum = cond(id[_n - 1] == "Empty" & id[_n - 2] == "Empty" & id[_n - 3] == "Empty" & id[_n - 4] ==
"Empty" & id[_n - 5] == "Empty" & id[_n - 6] == "Empty" & id[_n]== "Contact", pdfPageNum[_n-1]+1, pdfPageNum[_n-1
]) in 2/44754

* because the old combined register pdf only goes to 2022 I have to add on the most recent pages from the
currently available register on the fife website. Because its another pdf I have to have to restart pdf page number
replace pdfPageNum = 1 in 44755
replace pdfPageNum = cond(id[_n - 1] == "Empty" & id[_n - 2] == "Empty" & id[_n - 3] == "Empty" & id[_n - 4] ==
"Empty" & id[_n - 5] == "Empty" & id[_n - 6] == "Empty" & id[_n]== "Contact", pdfPageNum[_n-1]+1, pdfPageNum[_n-1
]) in 44756/l
replace pdfPageNum = . if id == "Empty" | id == "Contact"
* the consolidated split chunk is no longer a first row, update id to reflect this
replace id = "segment, first row, split chunk" if id == "First Row" & chunk_consolidator >= 2 & problem == "split
chunk"

// Must recreate sr_no chunk_num and rowwithinchunk to reflect consolidation of split chunks
rename sr_no old_sr_no
rename chunk_num old_chunk_num
rename rowwithinchunk rowwithinchunk_old

*creates new serial number
g sr_no_new = _n

* creates new chunk number variable
g new_chunk_num = .
replace new_chunk_num = 0 in 1
replace new_chunk_num = cond(id == "First Row", new_chunk_num[_n-1] + 1, new_chunk_num[_n-1]) in 2/l

* creates new row within chunk variable
gsort new_chunk_num sr_no_new
by new_chunk_num : gen rowwithinchunk_new  = _n

* identify the first row of each segment according to the decision data in the first row string
replace id = "segment, first row" if (regexm(var1, "(GRANT|CON|DEEM)$") & !(id == "First Row" | id == "segment,
first row, split chunk"))
replace id = "segment, first row" if id == "segment, first row, split chunk"
```

```
/*------------------------

EXTRACTING COMMALITIES IN THE SEGMENT FIRST ROWS TO DIFFERENTIATE BETWEEN HMOADRESS APPLICANTNAME AND AGENT NAME

------------------------*/
gen relevant_part = var1

* Manually apply the replace command for each status
replace relevant_part = regexs(1) if regexm(var1, "^(.*?)(Licence Expired)") & (id == "First Row" | id ==
"segment, first row")
replace relevant_part = regexs(1) if regexm(var1, "^(.*?)(Licence Superseded)") & (id == "First Row" | id ==
"segment, first row")
replace relevant_part = regexs(1) if regexm(var1, "^(.*?)(Licence Surrendered)") & (id == "First Row" | id ==
"segment, first row")
replace relevant_part = regexs(1) if regexm(var1, "^(.*?)(Licence Issued)") & (id == "First Row" | id ==
"segment, first row")
replace relevant_part = "" if id != "First Row" & id != "segment, first row"


gen relevant_less_appref = relevant_part

* Step 2: Identify and exclude the application reference number
replace relevant_less_appref = regexs(1) if regexm(relevant_part, "^(?:F0[^ ]+|F2[^ ]+) (.*)")
replace relevant_less_appref = "" if id != "First Row" & id != "segment, first row"

clear all

AT THIS POINT PYTHON COMMANDS IN LongestCommonSubstring.py WERE USED
```

```python
# import pandas
import pandas as pd
from difflib import SequenceMatcher

# function to find the longest common substring among a list of strings
# this is to fix the issue where no ward separates the address and applicant variables
def find_longest_common_substring(str_list):
    str_list = [str(x) for x in str_list]  # Convert all to strings in case of mixed types
    if not str_list:
        return ""
    common_substring = str_list[0]
    for s in str_list[1:]:
        matcher = SequenceMatcher(None, common_substring, s)
        match = matcher.find_longest_match(0, len(common_substring), 0, len(s))
        if match.size == 0:
            return ""  # No common substring found
        common_substring = common_substring[match.a: match.a + match.size]
    return common_substring

# import the csv of data exported from stata
file_path = "H:\\CSVCommon.csv"

# read the data into the pandas DataFrame
df = pd.read_csv(file_path)

# filter the DataFrame for rows where 'id' is "First Row" or "segment, first row"
filtered_df = df[(df['id'] == 'First Row') | (df['id'] == 'segment, first row')]

# filter the dataframe by 'new_chunk_num' and apply the function
grouped_substrings = filtered_df.groupby('new_chunk_num')['relevant_less_appref'].apply(find_longest_common_substring)

# create a new column 'similarity1' and map the longest common substrings to the original DataFrame,
#this will contain the address

df['similarity1'] = df['new_chunk_num'].map(grouped_substrings)

# ensure 'similarity1' is only populated for rows where 'id' is "First Row" or "segment, first row"
df['similarity1'] = df.apply(lambda row: row['similarity1']
                             if row['id'] in ['First Row', 'segment, first row'] else '', axis=1)

# Check the first few rows to ensure the mapping is correct
print(df.head())

# Save the updated DataFrame to a new CSV file if needed
df.to_csv('H:\\UpdatedCSVCommon.csv', index=False)

#this code can be used to test whether the code successfully extracted the address
# replace 'specific_chunk_number' with the actual new_chunk_num you want to inspect
specific_chunk_number = 1  # Example chunk number
print(df[df['new_chunk_num'] == specific_chunk_number][['new_chunk_num', 'similarity1']])
```

```
import delimited "C:\Users\forw1\OneDrive - University of St Andrews\Register\Data\UpdatedCSVCommon.csv", clear

//clear the address/similiar1 variable for licenceses with only 1 segment --> create segmentwithinno variable
* Sort the data by new_chunk_num to ensure proper grouping
sort new_chunk_num

* Create a flag that identifies the start of a new segment
gen segment_start = (id == "First Row" | id == "segment, first row")

* Generate the segmentwithinchunk variable that numbers each segment within a chunk
* This variable will incrementally number each segment within each new_chunk_num
bysort new_chunk_num: gen segmentwithinchunk = sum(segment_start)

* To make the segment numbering restart from 1 for each new_chunk_num
bysort new_chunk_num (segmentwithinchunk): replace segmentwithinchunk = segmentwithinchunk[_n] -
segmentwithinchunk[1] + 1 if _n > 1

* Optional: You might want to drop the temporary flag variable
drop segment_start

* Ensure the data is sorted by new_chunk_num, necessary for consistent group processing
sort new_chunk_num

* Create the segmentwithinchunkmax variable
egen segmentwithinchunkmax = max(segmentwithinchunk), by(new_chunk_num)
replace segmentwithinchunk = . if id == "Contact" | id == "Empty"
replace segmentwithinchunkmax = . if id == "Contact" | id == "Empty"

replace similarity1 = "" if id == "First Row" & segmentwithinchunkmax == 1
replace similarity1 = "" if id != "First Row"
// Deal wit the fact that the ward is included in many of the common strings
gen temp_pattern = "^" + ward

* Replace similarity1 with the version without WARD if WARD is contained in similarity1
replace similarity1 = regexr(similarity1, ward, "") if regexm(similarity1, ward)

* Drop the temporary pattern variable
drop temp_pattern
sort similarity1
```

AT THIS POINT PYTHON COMMANDS IN RemoveLastNameFromSimilarity1.py WERE USED

```python
import pandas as pd
import re

# Step 1: Read the CSV file
file_path = 'H:/SimilarityDataWLastNames.csv'  # Adjust the path as necessary
df = pd.read_csv(file_path)

# Step 2: Process the `similarity1` column
def update_similarity_string(s):
    # Check if the entry is not a string (e.g., NaN or None)
    if not isinstance(s, str):
        return s  # Return as is or handle accordingly
    # Check if the string starts with 'Flat', in which case return it unchanged
    if s.strip().startswith('Flat'):
        return s
    # Regex to match optional leading spaces, any word (not 'Flat') followed by spaces, and a 1-3 digit number
    pattern = r'^(?!Flat\b)\s*\b\w+\s+(\d{1,3})'
    match = re.search(pattern, s)
    if match:
        # Replace the string with the substring including and after the digit
        return s[match.start(1):].strip()
    return s

df['similarity1'] = df['similarity1'].apply(update_similarity_string)

# Step 3: Save the modified DataFrame
output_file_path = 'H:/SimilarityDataWLastNames_Modified.csv'  # Adjust the path as necessary
df.to_csv(output_file_path, index=False)

print("File has been processed and saved.")
```

```
import delimited "C:\Users\forw1\OneDrive - University of St
Andrews\Register\Data\SimilarityDataWLastNames_Modified.csv", clear

// cleaning up the data a bit/ replace straglers that broke the pattern
drop old_sr_no old_chunk_num rowwithinchunk_old
replace similarity1 = "Ladybrand 28 Lade " if apprefnumber1 == "F00344/12"
replace similarity1 = "Flat 4 Crawford House " if apprefnumber1 == "F00253/12"
replace similarity1 = "Flat 4 172 South Street " if apprefnumber1 == "F00169/12"
replace similarity1 = "The Flat" if apprefnumber1 == "F00425/12"
replace similarity1 = "Flat 4 172 South Street " if apprefnumber1 == "F01464/15"
replace similarity1 = "Flat 4 Crawford House " if apprefnumber1 == "F01473/15"
replace similarity1 = "Ladybrand 28 Lade " if apprefnumber1 == "F01635/15"
replace similarity1 = "The Flat " if apprefnumber1 == "F01727/15"
replace similarity1 = "Northgate 125 North" if apprefnumber1 == "F00157/12"


// deal with the "The University Of St Andrews" pattern breakers
g unidealer = ""
replace unidealer = regexs(1) if similarity1 == "The University Of St Andrews " & regexm(var1, "NW[0-9]{2} (.*?)(
Licence Expired| Licence Issued| Licence Superseded| Licence Surrendered|$)")

replace unidealer = regexs(1) if similarity1 == "The University Of St Andrews " & regexm(var1,"W5[A-Za-z0-9]{3}
(.*?)( Licence Expired| Licence Issued| Licence Superseded| Licence Surrendered|$)")
replace similarity1 = unidealer if similarity1 == "The University Of St Andrews "
drop unidealer

// deal with "University Court Of The University" pattern breakers
generate match = 0
replace match = 1 if substr(relevant_less_appref, 1, length(similarity1)) == similarity1
g unidealer = ""

* if match == 1 i.e. the address is in between the ward and the licence status
replace unidealer = regexs(1) if match == 1 & regexm(var1,"W5[A-Za-z0-9]{3} (.*?)( Licence Expired| Licence
Issued| Licence Superseded| Licence Surrendered|$)") & similarity1 == "University Court Of The University " |
similarity1 == "University Of St Andrews "
replace unidealer = regexs(1) if match == 1 & regexm(var1,"W5[A-Za-z0-9]{3} (.*?)( Licence Expired| Licence
Issued| Licence Superseded| Licence Surrendered|$)") & similarity1 == "University Court Of The University " |
similarity1 == "University Of St Andrews "
replace similarity1 = unidealer if match == 1 & similarity1 == "University Court Of The University " | similarity1
 == "University Of St Andrews "

* if match == 0 i.e. the address is in between the the ward and the "University Court Of The University"
replace unidealer = substr(var1, strpos(var1, "W5R18") + length("W5R18"), ///
strpos(var1, "University Court Of The University ") - strpos(var1, "W5R18") - length("W5R18")) ///
if similarity1 == "University Court Of The University " & match == 0
replace similarity1 = unidealer if similarity1 == "University Court Of The University "
```

```
// deal with the "a", "an" "e", "er", "n", "on", "y" exceptions

* a pattern breaker
replace unidealer = substr(var1, strpos(var1, ward) + length(ward), .) if similarity1 == "a"
replace similarity1 = unidealer if similarity1 == "a"

* an e n y
*replace unidealer = regexs(2) if regexm(similarity1, "^(an |e |n |y )\S+\s+(.*)")
replace unidealer = regexs(2) if regexm(similarity1, "^(an |e |n |y )\S+\s+(.*)") & _n >= 48066 & _n <= 48092
g tempvar = 1 if regexm(similarity1, "^(an |e |n |y )\S+\s+(.*)") & _n >= 48066 & _n <= 48092
replace similarity1 = unidealer if tempvar == 1
drop tempvar
sort similarity1 // looking good!

// deal with the segmentwithinchunkmax == 1 exceptions & ward is not empty
br if id == "First Row" & segmentwithinchunkmax == 1 & ward != ""
* if there is a ward number there is the HMOAddress and applicant name can be clearly differentiated
APPLICANTNAME WARD HMOADDRESS
replace similarity1 = substr(relevant_less_appref, strpos(relevant_less_appref, ward) + length(ward), .) if id ==
"First Row" & segmentwithinchunkmax == 1 & ward != ""

// deal with the segmentwithinchunkmax == 1 exceptions & ward is empty
br if id == "First Row" & segmentwithinchunkmax == 1 & ward == ""

* extract everything after the word "Flat"
generate temp_flat_pos = strpos(relevant_less_appref, "Flat") if id == "First Row" & segmentwithinchunkmax == 1 &
ward == ""
replace similarity1 = substr(relevant_less_appref, temp_flat_pos, .) if temp_flat_pos > 0 & segmentwithinchunkmax
== 1 & id == "First Row" & ward == ""

* if there is a number, and "Flat" isnt contained in the string extract everything after the number as the
"similarity1" variable
replace similarity1 = regexs(1) if regexm(relevant_less_appref, "([0-9].*)") & strpos(relevant_less_appref, "Flat"
) == 0 & id == "First Row" & segmentwithinchunkmax == 1 & ward == ""

br if id == "First Row" & segmentwithinchunkmax == 1 & ward == "" & similarity1 == ""
br if similarity1 == "" & id == "First Row"
// Now that all addresses are extracted in similarity1, the applicant name can be extracted even from those
string that do not have the ward

sort similarity1 sr_no_new

br if ward == "" & id == "First Row"

generate ApplicantName = ""

* Extract the part of relevant_less_appref that comes before the string in similarity1
replace ApplicantName = substr(relevant_less_appref, 1, strpos(relevant_less_appref, similarity1) - 1) if ward ==
"" & id == "First Row" & strpos(relevant_less_appref, similarity1) > 0 & similarity1 != ""
replace ApplicantName = "" if apprefnumber1 == "F02828/18" |apprefnumber1 == "F02815/18"| apprefnumber1 ==
"F02713/17"

br if ward != "" & id == "First Row"

* extract address if there is a ward to distinguish the information
gen ward_atstart = 0
replace ward_atstart = 1 if strpos(relevant_less_appref, ward) == 1 & ward != "" & id == "First Row"
replace ApplicantName = substr(relevant_less_appref, 1, strpos(relevant_less_appref, ward) - 1) if ward_atstart !=
 1 & ward != "" & id == "First Row" & strpos(relevant_less_appref, ward) > 0

count if id == "First Row" & ApplicantName == "" // 196
count if ward_atstart == 1 // 148
```

```
/*-----------------------------------
|                                   |
|     Creating Expedited Code       |
|                                   |
|---------------------------------*/

// Extract Date variables + Occupants Variable
gen AppDate = regexs(2) if id == "First Row" & regexm(var1, "(Licence Expired|Licence Issued|Licence
Superseded|Licence Surrendered) (\d{2}/\d{2}/\d{4})")

gen IssueDate = regexs(1) if id == "First Row" & regexm(var1, "\d{2}/\d{2}/\d{4} (\d{2}/\d{2}/\d{4})
\d{2}/\d{2}/\d{4}")

gen ExpireDate = regexs(1) if regexm(var1, ".*(?:\d{2}/\d{2}/\d{4}).*(\d{2}/\d{2}/\d{4})\s\d{1,3}") & id ==
"First Row"

gen totOccs = regexs(1) if regexm(var1, "\d{2}/\d{2}/\d{4} (\d{1,3}) (CON|GRANT|DEEM)") & id == "First Row"
destring totOccs, replace

// See if there are pattern breakers
count if IssueDate == "" & id == "First Row" // 65
replace IssueDate = ExpireDate if IssueDate == "" & id == "First Row"

count if IssueDate == "" & id == "First Row" // now 10
count if IssueDate == "" & id == "First Row" & ExpireDate == "" // 10 (same cases as issue date empty)

replace IssueDate = "06/07/2017" if apprefnumber1 == "F02037/16"
replace ExpireDate = "16/09/2019" if apprefnumber1 == "F02037/16"
replace totOccs = 3 if apprefnumber1 == "F02037/16"

replace IssueDate = "24/10/2017" if apprefnumber1 == "F01899/16"
replace ExpireDate = "03/06/2019" if apprefnumber1 == "F01899/16"
replace totOccs = 4 if apprefnumber1 == "F01899/16"

* this is is from kirkaldy (no reason to fix) replace IssueDate = "03/11/2017" if apprefnumber1 == "F01900/16"
* if apprefnumber1 == "F01900/16"
replace IssueDate = "22/02/2023" if apprefnumber1 == "F04504/22"
replace ExpireDate = "25/05/2025" if apprefnumber1 == "F04504/22"
replace totOccs = 3 if apprefnumber1 == "F04504/22"

* + one random from glenrothes
replace IssueDate = "17/08/2018" if apprefnumber1 == "F02815/18"
replace ExpireDate = "23/05/2021" if apprefnumber1 == "F02815/18"
replace totOccs = 6 if apprefnumber1 == "F02815/18"

replace IssueDate = "05/12/2018" if apprefnumber1 == "F02713/17"
replace ExpireDate = "19/01/2021" if apprefnumber1 == "F02713/17"
replace totOccs = 3 if apprefnumber1 == "F02713/17"

replace IssueDate = "04/10/2018" if apprefnumber1 == "F02828/18"
replace ExpireDate = "04/10/2021" if apprefnumber1 == "F02828/18"
replace totOccs = 4 if apprefnumber1 == "F02828/18"

replace IssueDate = "27/11/2014" if apprefnumber1 == "F00827/14"
replace ExpireDate = "21/03/2017" if apprefnumber1 == "F00827/14"
replace totOccs = 3 if apprefnumber1 == "F00827/14"

replace IssueDate = "20/12/2016" if apprefnumber1 == "F02053/16"
replace ExpireDate = "18/10/2019" if apprefnumber1 == "F02053/16"
replace totOccs = 3 if apprefnumber1 == "F02053/16"
// Selecting only properties in St. Andrews
gen Andrews = regexm(var1, "(?i)Andrews")

sort new_chunk_num sr_no_new

* Create a flag variable for the first row in each chunk
by new_chunk_num: gen firstRow = (_n == 1)

* Propagate the Andrews value to the first row if any of the next three rows within the same chunk have Andrews
equal to 1
by new_chunk_num: replace Andrews = 1 if (Andrews[_n+1] == 1 | Andrews[_n+2] == 1 | Andrews[_n+3] == 1 |Andrews[_n
+4] == 1) & firstRow

count if Andrews != 1 & firstRow == 1 & ward != "W5R18"

count if Andrews == 1 & id == "First Row" &( IssueDate == "" | ExpireDate == "") // 0 no properties marked as in
St, Andrews are without IssueDate or ExpireDate variables

drop if id != "First Row"
drop if Andrews != 1
```

```
/*----------------------------------
|                                  |
|     Section 3: Transformation    |
|                                  |
|----------------------------------*/
/*
        Note: code copied/modified from Week9_ListDataOptimal_FW (apologies for loops)
*/
br

gen IssueDate_new = date(IssueDate, "DMY")
gen ExpireDate_new = date(ExpireDate, "DMY")


cd "`data'"
save ExpeditedPDFData2012, replace
* Now you have a dataset where each licence has a row for each month in the specified range,
* and the Active variable indicates whether the licence was active in that month.

clear
set obs 1
gen month = 1
expand 12 * (2026 - 2012 + 1)
bysort month: gen year = 2012 + floor((_n-1)/12)
replace month = mod(_n-1, 12) + 1
gen ym = ym(year, month)
gen ym_date = dofm(ym)
tempfile monthlist
save "`monthlist'"

* Step 2: Load original dataset and cross join with the month list
use "ExpeditedPDFData2012.dta"  // replace with the path to your dataset

tempfile licenses
save "`licenses'"
cross using "`monthlist'"

* Step 3: Calculate the 'Active' variable for each license for each month
gen Active = (IssueDate_new <= ym_date) & (ym_date <= ExpireDate_new)
*replace Active = cond(Active, 1, 0)

* The dataset now has a row for each license for each month from 2018 to 2026 with an 'Active' flag
sort apprefnumber1 ym_date  // replace 'license_id' with your actual license identifier variable

save ExpeditedData_2012Onwards, replace
```

Appendix 9: Code Creating Active Monthly Licences Graph

```
// Set Directory
cd "`data'"

// Import "Kategorical" Dataset
use "`data'FOIRegister_Cleaned"

// drop observations to include only variables of interest
drop if year < 2011
drop if year > 2023

* Generate a variable for total active occupants
gen totalActive = Active

* Create a date variable that Stata can recognize
gen date = ym(year, month)

* Format the date variable to display as a date
format date %tm

* Aggregate the data to get the total of active occupants by month
collapse (sum) totalActive, by(date)

tsset date
tsline totalActive, title("Active Licences from 2011 to 2023") ///
    ytitle("Number of Active Licences") xtitle("Time") ///
    ylabel(500(100)1200, grid) xlabel(612(12)767, angle(45) format(%tm) labsize(small)) ///
    graphregion(color(white)) xmtick(#20, grid)
```

Appendix 10: Code Creating Smoothed Monthly Active Licences Graph

```
// Import "Kategorical" Dataset
use "`data'FOIRegister_Cleaned"

// drop observations to include only variables of interest
drop if year < 2011
drop if year > 2023

* Generate a variable for total active occupants
gen totalActive = Active

* Create a date variable that Stata can recognize (assuming Month and Year are numeric)
gen date = ym(year, month)

* Format the date variable to display as a date
format date %tm

* Aggregate the data to get the total of active occupants by month
collapse (sum) totalActive, by(date)

* Smoothing
tsset date
tssmooth ma totalActive_smoothed = totalActive, window(12 1)

export excel using "`output'MonthlyActiveLicenses_PR.xlsx", replace firstrow(variables)

* Create the monthly graph of total active occupants
tsset date
tsline totalActive_smoothed, title("Active Licences from 2011 to 2024 (Moving Average)") ///
    ytitle("Number of Active Licences") xtitle("Time") ///
    ylabel(500(100)1200, grid) xlabel(612(12)767, angle(45) format(%tm) labsize(small)) graphregion(color(white))
xmtick(#20, grid) ///
    subtitle("Twelfth order moving average, Uni halls removed")
```

Appendix 11: Code Creating Active Bedrooms by Month Graph

```
// Set Directory
cd "`data'"

// Import "Kategorical" Dataset
use "`data'FOIRegister_Cleaned"

* drop if TotOccs > 10
// drop observations to include only variables of interest
drop if year < 2011
drop if year > 2023

* Generate a variable for total active occupants
gen totalActiveOccs = TotOccs * Active

* Create a date variable that Stata can recognize (assuming Month and Year are numeric)
gen date = ym(year, month)
*replace date = dofm(date)
* Format the date variable to display as a date
format date %tm

* Aggregate the data to get the total of active occupants by month
collapse (sum) totalActiveOccs, by(date)

* Create the monthly graph of total active occupants
tsset date
tsline totalActiveOccs, title("Monthly Total of Active Occupants from 2011 to 2024") ///
    ytitle("Total Active Occupants") xtitle("Time") ///
    ylabel(3000(500)7500, grid) xlabel(612(12)767, angle(45) format(%tm) labsize(small)) graphregion(color(white))
xmtick(#20, grid) ///
```

Appendix 12: Code Creating Smoothed Active Bedrooms by Month Graph

```
// Import "Kategorical" Dataset
use "`data'FOIRegister_Cleaned"

* Generate a variable for total active occupants
gen totalActiveOccs = Active * TotOccs


* Create a date variable that Stata can recognize (assuming Month and Year are numeric)
gen date = ym(year, month)

* Format the date variable to display as a date
format date %tm

* Drop observations to include only variables of interest
drop if year < 2011
drop if year > 2023

* Aggregate the data to get the total of active occupants by month
collapse (sum) totalActive, by(date)

* Smoothing
tsset date
tssmooth ma totalActive_smoothed = totalActiveOccs, window(12 1)

export excel using "`output'MonthlyActiveBedroom_PR.xlsx", replace firstrow(variables)

tsset date
tsline totalActive_smoothed, title("Monthly Total of Active Occupants from 2011 to 2024") ///
    ytitle("Total Active Occupants") xtitle("Time") ///
    ylabel(3000(500)7000, grid) xlabel(612(12)767, angle(45) format(%tm) labsize(small)) graphregion(color(white))
xmtick(#20, grid) ///
    subtitle("Twelfth order moving average")
```

Appendix 13: BedroomsAsPropofTotProp_Revised.do

```
drop if year < 2010
drop if year > 2023

/// Attempting to recreate the quarterly graph

gen Active_3Bed = 0
replace Active_3Bed = 1 if Active == 1 & TotOccs == 3

gen Active_4Bed = 0
replace Active_4Bed = 1 if Active == 1 & TotOccs == 4

gen Active_5Bed = 0
replace Active_5Bed = 1 if Active == 1 & TotOccs == 5

gen Active_6Bed = 0
replace Active_6Bed = 1 if Active == 1 & TotOccs >= 6

forval year = 2010/2023{
    forval month = 1/12{
        forval bedtype = 3/6{
            count if Active_`bedtype'Bed == 1 & month == `month' & year == `year'
            scalar Total`bedtype'_`month'_`year' = r(N)
            display "Total 6 bed = " Total`bedtype'_`month'_`year'
        }
        count if Active == 1 & month == `month' & year == `year'
        scalar totalActive_`month'_`year' = r(N)

        scalar prop3bed`month'`year' = Total3_`month'_`year' / totalActive_`month'_`year'
        scalar prop4bed`month'`year' = Total4_`month'_`year' / totalActive_`month'_`year'
        scalar prop5bed`month'`year' = Total5_`month'_`year' / totalActive_`month'_`year'
        scalar prop6bed`month'`year' = Total6_`month'_`year' / totalActive_`month'_`year'
    }
}

clear
set obs 168
gen date = ym(2010,1) + _n -1
gen daily_date = dofm(date)
format daily_date %tdDD/NN/YYYY

gen value3 = .
gen value4 = .
gen value5 = .
gen value6 = .

gen date_fr = date("`daily_date'", "DMY")

// Create time and value variables for graph
local i = 1
forval year = 2010/2023 {
    forval month = 1/12 {
        forval bedtype = 3/6 {
            replace value`bedtype' = prop`bedtype'bed`month'`year' in `i'
        }
        local i = `i' + 1
    }
}
}
// Create graph
twoway (line value3 date, lc(red) lw(thin)) || ///
       (line value4 date, lc(blue) lw(thin)) || ///
       (line value5 date, lc(green) lw(thin)) || ///
       (line value6 date, lc(purple) lw(thin)), ///
       legend(label(1 "3 Bedrooms") label(2 "4 Bedrooms") label(3 "5 Bedrooms") label(4 "6+ Bedrooms")) ///
       xlab(600(9)767, angle(60) format(%tmM-Y) labsize(small)) xtick(600(9)767) ylab(0(.1).6) ytick(0(.1).6)
 ///
       title("Bedroom Type as a Proportion of Total Active Licenses") xtitle("Month") ytitle("Proportion")
text(0.8 711 " Policy Enacted", placement(right) color(black)) text(0.8 757 "Policy Review  ", placement(left
) color(black))
```

Appendix 14: Rent Data Cleaning and Analysis

```
// Import data
import excel "`data'/Private Rental Information 2021 onwards", sheet("Data") firstrow clear


*correcting Town data - some typo mistakes generating more than one category for the same Town*

*make all the names the same - there was "St. Andrews" and "St Andrews"

replace Town = "St. Andrews" if Town == "St Andrews"
replace Town = "Balmullo" if Town == "Bulmullo"
replace Town = "Methil" if inlist(Town, "Methil ", "Methi", "Methilhill")
replace Town = "St. Monans" if Town == "St Monans"
replace Town = "Thornton" if Town == "THornton"
replace Town = "Townhill" if Town == "Townhill "
replace Town = "Kinghorn" if Town == "Kinghorn "
*same for some Letting Agencies
replace LettingAgent = "Zenlet" if LettingAgent == "ZenLet"
replace LettingAgent = "Professional Property Lettings" if LettingAgent == "Professional Propertay Ltd"
*same for County
replace County = "Fife" if County == "Fife "

replace NoofBedrooms = "6" if NoofBedrooms == "6+"
destring(NoofBedrooms), replace

** RentPCM has an outlier (12000) - found similar property online that seems to be a castle-like house that can
hosts events... not accurate representation of rental properties => trim from data
drop if RentPCM == 12000

** save total number of observations as a variable to use in title of histogram
*** for the moment i havent found a way to save this so that it works for all the histogramsin the do file so it
needs to be put before each histogram tht will include the total frequency in the title
scalar total_obs = _N
local total_obs_str = string(total_obs)
hist RentPCM, bin(50) title("Distribution of Rent PCM (in £) (Total frequency: `total_obs_str') " )
graph export "`output'rent_hist.png", as(png) replace
** distribution of rent PCM


**distribution of Noofbedrooms
scalar total_obs = _N
local total_obs_str = string(total_obs)
graph pie, over(NoofBedrooms) plabel(_all percent) title("Proportion of Number of bedrooms (Total frequency:
`total_obs_str') ")

graph export "`output'bedrooms_hist.png", as(png) replace
**comparing rent PCM to number of bedrooms
hist RentPCM, by(NoofBedrooms) bin(10) title("Rent PCM  ")
graph export "`output'rent_beds_hist.png", as(png) replace

**generating new variable rent per person to compare how rent changes per person depending on number of bedrooms*
gen RentPP = RentPCM / max(NoofBedrooms, 1)
label var RentPP "Rent Per Person"

**distribution of rent pp (assuming max 1 person per bedroom)
scalar total_obs = _N
local total_obs_str = string(total_obs)
hist RentPP, bin(30) title("Distribution of Rent Per Person (in £) Total frequency: `total_obs_str')")
graph export "`output'rentPP_hist.png", as(png) replace

** comparing Rent PP to number of bedrooms
hist RentPP, by(NoofBedrooms) bin(10) title("Rent PP ")
graph export "`output'rentpp_beds_hist.png", as(png) replace


** Replace town names with fewer than 10 observations to "other"
levelsof Town, local(Townlists)
foreach Town of local Townlists  {
    summarize NoofProperties if Town == "`Town'"
    scalar total_frequency = r(N)
    local total_frequency_str = string(total_frequency)
    replace Town = "Other" if Town == "`Town'" & `total_frequency_str' <= 14
}

*** proportion of properties per Town
scalar total_obs = _N
local total_obs_str = string(total_obs)
graph pie, over(Town) sort descending plabel(_all percent) title("Proportion of properties per Town") subtitle(
"Total frequency: `total_obs_str'")
graph export "`output'towns_pie_chart.png", as(png) replace
```

```
**Top 3 Towns (between 85-115 obs): Kirkcaldy, St. Andrews, Dunfermline
tab Town, sort

hist RentPCM if Town == "St. Andrews" | Town == "Kirkcaldy" | Town == "Other" | Town == "Dunfermline", by(Town)
bin(10) title("Rent PCM in ")
graph export "`output'rent_alltown_hist.png", as(png) replace

*** proportion of properties per HMA
scalar total_obs = _N
local total_obs_str = string(total_obs)
graph pie, over(HMA) sort descending plabel(_all percent) title("Proportion of properties per HMA") subtitle(
"Total frequency: `total_obs_str'")
graph export "`output'HMA_pie_chart.png", as(png) replace

**rent PCM in each HMA
hist RentPCM, by(HMA) bin(10) title("Rent PCM in ")
graph export "`output'rent_allHMA_hist.png", as(png) replace


** Replace propertytype with fewer than 20 observations to "other"
levelsof PropertyType, local(PropertyTypelist)
foreach PropertyType of local PropertyTypelist {
    summarize NoofProperties if PropertyType == "`PropertyType'"
    scalar total_frequency = r(N)
    local total_frequency_str = string(total_frequency)
    replace PropertyType = "Other" if PropertyType == "`PropertyType'" & `total_frequency_str' <= 20
}
** Property type pie chart
scalar total_obs = _N
local total_obs_str = string(total_obs)
graph pie, over(PropertyType) sort descending plabel(_all percent) title("Property Types (Total frequency:
`total_obs_str') ") subtitle("Proportion of Each Category")
graph export "`output'property_type_pie_chart.png", as(png) replace


**rent PCM distribution for each propertytype
hist RentPCM, by(PropertyType) bin(10) title("Rent PCM for ")
graph export "`output'rent_allPropertyType_hist.png", as(png) replace


** rent PCM for Top 3 LettingAgent in St. Andrews (10 or more properties in data)
tab LettingAgent if Town == "St. Andrews", sort

** Replace lettingagent with fewer than 10 observations to "other"
levelsof LettingAgent, local(LettingAgentlists)
foreach LettingAgent of local LettingAgentlists {
    summarize NoofProperties if LettingAgent == "`LettingAgent'" & Town == "St. Andrews"
    scalar total_frequency = r(N)
    local total_frequency_str = string(total_frequency)
    replace LettingAgent = "Other" if LettingAgent == "`LettingAgent'" & Town == "St. Andrews" &
`total_frequency_str' <= 6
}


** LettingAgent pie chart
scalar total_obs = _N
local total_obs_str = string(total_obs)
graph pie, over(LettingAgent) sort descending plabel(_all percent) title("Letting Agencies in St Andrews (Total
frequency: `total_obs_str') ") subtitle("Proportion of Each Category")
graph export "`output'all_lettingAgent_pie_chart.png", as(png) replace


** LettingAgent pie chart for st andrews
summarize NoofProperties if Town == "St. Andrews"
scalar total_frequency = r(N)
local total_frequency_str = string(total_frequency)
graph pie if Town == "St. Andrews", over(LettingAgent) sort descending plabel(_all percent) title("Letting
Agencies in St Andrews (Total frequency: `total_frequency_str') ") subtitle("Proportion of Each Category")
graph export "`output'lettingAgent_pie_chart.png", as(png) replace


** rent PCM for Top 5 LettingAgent in St. Andrews (7 or more properties in data)
hist RentPCM if  Town == "St. Andrews", sort by(LettingAgent) bin(5) title("Rent PCM for Agency: ")
graph export "`output'rent_alllettings_hist.png", as(png) replace


** LHS Area pie chart
scalar total_obs = _N
local total_obs_str = string(total_obs)
graph pie, over(LHSArea) sort descending plabel(_all percent) title("Local Housing Strategy Areas (Total
frequency: `total_obs_str') ") subtitle("Proportion of Each Category")
graph export "`output'LHSArea_pie_chart.png", as(png) replace

** rent PCM for *Top 4: Dunfermline & the Coast, Kirkcaldy, St. Andrews & Levenmouth
hist RentPCM if LHSArea == "Dunfermline & the Coast" | LHSArea == "Kirkcaldy" | LHSArea == "St. Andrews" | LHSArea
 == "Levenmouth", sort by(LHSArea) bin(10) title("Rent PCM in LHS Area: ")
graph export "`output'rent_all_LHSArea_hist.png", as(png) replace
```

```
** Furnished vs Unfurnished
scalar total_obs = _N
local total_obs_str = string(total_obs)
graph pie, over(FurnishedUnfurnished) plabel(_all percent) title("Furnishing Status in Fife Properties Total
frequency: `total_obs_str'") subtitle("Proportion of Each Category")
graph export "`output'furnishing_pie_chart.png", as(png) replace

** Furnished vs Unfurnished in St Andrews
summarize NoofProperties if Town == "St. Andrews"
scalar total_frequency = r(N)
local total_frequency_str = string(total_frequency)
graph pie if Town == "St. Andrews", over(FurnishedUnfurnished) plabel(_all percent) title("Furnishing Status in
St. Andrews properties ") subtitle("Total frequency: `total_frequency_str'")
graph export "`output'furnishing_sta_pie_chart.png", as(png) replace

** rent PCM for furnished/unfurnished properties
hist RentPCM, sort by(FurnishedUnfurnished) bin(20) title("Rent PCM for Properties that are ")
graph export "`output'rent_allFurnishedUnfurnished_hist.png", as(png) replace


** proportion of properties from each academic year
scalar total_obs = _N
local total_obs_str = string(total_obs)
graph pie, over(Year) plabel(_all percent) title("Properties from each academic year ") subtitle("Total
frequency: `total_obs_str'")
graph export "`output'years_pie_chart.png", as(png) replace

** rent PCM distribution for different academic years
hist RentPCM, sort by(Year) bin(20) title("Rent PCM in ")
graph export "`output'rent_all_years_hist.png", as(png) replace

* furnish status by number of bedrooms
graph pie, by(NoofBedrooms) over(FurnishedUnfurnished) plabel(_all percent)
graph export "`output'beds_furnish_pie_chart.png", as(png) replace

* number of bedrooms by year
graph pie, by(Year) over(NoofBedrooms)
graph export "`output'years_beds_pie_chart.png", as(png) replace

* property type by number of bedrooms
graph pie, by(NoofBedrooms) over(PropertyType)
graph export "`output'beds_propertytype_pie_chart.png", as(png) replace

* number of bedrooms by Town
graph pie if Town == "St. Andrews" | Town == "Kirkcaldy" | Town == "Other" | Town == "Dunfermline", by(Town) over(
NoofBedrooms)
graph export "`output'towns_beds_pie_chart.png", as(png) replace

* propertytype by year
graph pie, by(Year) over(PropertyType)
graph export "`output'years_propertytype_pie_chart.png", as(png) replace

* PropertyType by Town
graph pie if Town == "St. Andrews" | Town == "Kirkcaldy" | Town == "Other" | Town == "Dunfermline", by(Town) over(
PropertyType)
graph export "`output'towns_propertytype_pie_chart.png", as(png) replace

* propertytype by letting agents in st andrews
graph pie if Town == "St. Andrews", by(LettingAgent) over(PropertyType)
graph export "`output'lettings_propertytype_pie_chart.png", as(png) replace

*generating variables for regression
gen StAndrewsTown = 0
replace StAndrewsTown = 1 if Town == "St. Andrews"

gen FurnishStatus = 0
replace FurnishStatus = 1 if FurnishedUnfurnished == "Furnished"
replace FurnishStatus = 0.5 if FurnishedUnfurnished == "Part Furnished"

gen DunfermlineWF = 0
replace DunfermlineWF = 1 if HMA == "Dunfermline & West Fife"
gen KirkcaldyCF = 0
replace KirkcaldyCF = 1 if HMA == "Kirkcaldy & Central Fife"
gen StAndrewsNEF = 0
replace StAndrewsNEF = 1 if HMA == "St. Andrews & North East Fife"
gen CuparNW = 0
replace CuparNW = 1 if HMA == "Cupar & North West Fife"

gen Flat = 0
replace Flat = 1 if PropertyType == "Flat"

gen Year21_22 = 0
replace Year21_22 = 1 if Year == "2021-22"
gen Year23_24 = 0
replace Year23_24 = 1 if Year == "2023-24"

*initial regression

regress RentPCM NoofBedrooms StAndrewsTown FurnishStatus KirkcaldyCF CuparNW Flat Year21_22 Year23_24, vce(robust)

regress RentPCM NoofBedrooms StAndrewsTown FurnishStatus KirkcaldyCF CuparNW Flat Year21_22 Year23_24
```

Appendix 15 : Zoopla Scraping Code : Python

```
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"]

#separate 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 = []
print(link_list)

#go through links in link list from previous part
try:
    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, index=[0])
            #change Date from string to datetime
            df['Date'] = pd.to_datetime(df['Date'], format='%Y%m%d')
            #delete duplicate
            df = df.drop_duplicates()
            df=(df.T)
            print(df)
            #change the format to csv
            df.to_excel('housinginfo.xlsx')

except:
    print("connection error")
```

Appendix 16: **HousesForSaleToRent web scraping code (Python)**

```
from bs4 import BeautifulSoup
import re
```

```
import requests
import pandas as pd
import urllib.request
# import urllib.parse
# url_mapping = https://nominatim.openstreetmap.org/ui/search.html


#set up headers
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"}

#set up lists for data and parameter "number of page"
housinginfo_list = []
number_of_page = 31

#create function to scrape housing information from website
def HouseToRent(page):
  #url of renting webstie
  url_rent_stAndrews = f"https://housesforsaletorent.co.uk/houses/to-
rent/fife/saint-andrews.html?page={page}"
  #scrape using beautifulsoup
  r = requests.get(url_rent_stAndrews,headers=headers)
  soup = BeautifulSoup(r.text, "html.parser")
  results = soup.find_all("div", class_="col-4 justify listing-block")

  for item in results:
    location = item.find('h4',itemprop="address").text.strip()
    if "St Andrews" in location or "St. Andrews" in location:
      if "KY16" in location or "KY 16" in location:
        rent = item.find('div',class_ = 'price-info').text.strip()
        #code in green intends to get longitude and lattitude of location
for geographical mapping in the next stage
        # location_for_mapping = location.replace(', KY16','')
        # response = requests.get(url_mapping).json()
        Housing = {
        'location': location,
        'room_number': item.find('h3', itemprop = "name").text[:1],
        'rent': ''.join(filter(str.isdigit, rent)),
        'link': f'https://housesforsaletorent.co.uk/houses/to-
rent/fife/saint-andrews.html?page={page}',
        # 'lat': response[0]["lat"],
        # 'lon': response[0]["lon"],
        }
        housinginfo_list.append(Housing)
  return housinginfo_list

#scraping each page on the website
for i in range(1,number_of_page+1):
  HouseToRent(i)

#create data frame to store obtained data
df = pd.DataFrame(housinginfo_list)

#remove duplicate rows
df = df.drop_duplicates()
print(df)

#save the scraping result into excel file
df.to_csv('housinginfo.csv')
print('housinginfo.csv')
```

Appendix Y1: Code to generate Price Index (Supply Team)

```
reg AdjustedRentPP yearsince2012 CentredDistance_weighted RoomDummy* [
aweight=Rooms],  noconstant


** Calculating weighted sums of coefficients for base year, 2012
local Rooms 1 2 3 4 5 6
*set sum of weights and p_2012 to zero
scalar sum_weights = 0
scalar p_2012_adj = 0

* Loop through each room dummy variable
forval i = 1/6 {

        * Calculating proportion for the ith RoomDummy
        sum RoomDummy`i'
        scalar proportion_RoomDummy`i' = r(sum) / _N
        * Calculating weight for the ith RoomDummy
        scalar weights_RoomDummy`i' = proportion_RoomDummy`i' * `i'
        * Update sum of weights
        scalar sum_weights = sum_weights + weights_RoomDummy`i'
        * Save coefficients from regression
        scalar coef_RoomDummy`i' = _b[RoomDummy`i']
        * Update p_2012_adj with weighted coeffs
        scalar p_2012_adj = p_2012_adj + (coef_RoomDummy`i' *
weights_RoomDummy`i')

    }

**divide by sum of weights
scalar p_2012_adj = p_2012_adj / sum_weights

gen p_year_adj1 = (p_2012_adj + _b[yearsince2012] * yearsince2012)

twoway (line p_year_adj1 Year), title("Price Index Adjusted") ytitle(
"Rent per person") xtitle("Year")

scatter p_year_adj1 Year
```

Appendix Y2: Code to generate variable measuring weighted distance to University Sports Centre

```
*Location measure of sports centre(56.34076737141038, -2.8098889841424266)
generate gymdist = sqrt((56.34076737141038 - Latitude)^2 + (-2.8098889841424266 -
Longitude)^2)
generate gymdist_km = gymdist*110

** Calculating mean of DistancetoTown
summarize gymdist_km, meanonly
global meanGymDist = r(mean)

** Calculating the weighted centred distance to the sports centre
* Calculate the weighted mean distance to the sports centre
sum gymdist_km [aweight=Rooms]
scalar weighted_mean_distance_gym = r(mean)
* Calculate CentredDistance based on the weighted mean
gen CentredDistancegym_weighted = gymdist_km - weighted_mean_distance_gym
```

Appendix 19: Demand team linear model codes

```
import numpy as np
from sympy import *
from sympy import symbols
from scipy.optimize import fsolve
import matplotlib.pyplot as plt

#defining symbols
elast2016,elast2017,elast2018,elast2019,
elast2020,elast2022,elast2023,elasticity_in = symbols("\
elast2016 elast2017 elast2018 elast2019 elast2020 elast2022 elast2023
elasticity_in")
#defining more symbols
A2020,B2020,H2016,H2017,H2018,H2019,H2020,H2022,H2023,StNH2016,StNH2017,StNH20
18,StNH2019,StNH2020,StNH2022,StNH2023 = symbols("\
A2020 B2020 H2016 H2017 H2018 H2019 H2020 H2022 H2023 StNH2016 StNH2017
StNH2018 StNH2019 StNH2020 StNH2022 StNH2023")
#defining more symbols
p2016,p2017,p2018,p2019,p2020,p2022,p2023,A2018,B2018,A2022,B2022,A2016,B2016,
A2017,B2017,A2019,B2019,A2023,B2023 = symbols("\
p2016 p2017 p2018 p2019 p2020 p2022 p2023 A2018 B2018 A2022 B2022 A2016 B2016
A2017 B2017 A2019 B2019 A2023 B2023")

Monthlyrent2016=554
Monthlyrent2017=535
Monthlyrent2018=644
Monthlyrent2019=661
Monthlyrent2020=640
Monthlyrent2022=677
Monthlyrent2023=676
EQNS=[]   ## initialise an empty array of equations
#Equations for demand
#These are general equations with A, B as unknowns and p as the set of all
prices
def xD2016(p):
   return A2016-B2016*p
def xD2017(p):
   return A2017-B2017*p
def xD2018(p):
   return A2018-B2018*p
def xD2019(p):
   return A2019-B2019*p
def xD2020(p):
   return A2020-B2020*p
def xD2022(p):
   return A2022-B2022*p
def xD2023(p):
   return A2023-B2023*p
```

```
### Input parameters
# Total supply of HMO rooms (from register data)
EQNS.append(H2016 - 2764)
EQNS.append(H2017 - 2879)
EQNS.append(H2018 - 2758)
EQNS.append(H2019 - 2718)
EQNS.append(H2020 - 2718)
EQNS.append(H2022 - 2718)
EQNS.append(H2023 - 2718)
# Student population not in halls  (from Uni FOI data)
EQNS.append(StNH2016 - 5003)
EQNS.append(StNH2017 - 5343)
EQNS.append(StNH2018 - 5071)
EQNS.append(StNH2019 - 5175)
EQNS.append(StNH2020 - 6381)
EQNS.append(StNH2022 - 6031)
EQNS.append(StNH2023 - 6000)
# Price observations (from price team data)
EQNS.append(p2016 - Monthlyrent2016)
EQNS.append(p2017 - Monthlyrent2017)
EQNS.append(p2018 - Monthlyrent2018)
EQNS.append(p2019 - Monthlyrent2019)
EQNS.append(p2020 - Monthlyrent2020)
EQNS.append(p2022 - Monthlyrent2022)
EQNS.append(p2023 - Monthlyrent2023)
## elasticity assumptions
EQNS.append(elast2016 - elasticity_in)
EQNS.append(elast2017 - elasticity_in)
EQNS.append(elast2018 - elasticity_in)
EQNS.append(elast2019 - elasticity_in)
EQNS.append(elast2020 - elasticity_in)
EQNS.append(elast2022 - elasticity_in)
EQNS.append(elast2023 - elasticity_in)
#### Elasticities evaluated at eqm prices
EQNS.append(elast2016+diff(xD2016(p2016), p2016)*p2016/xD2016(p2016))
EQNS.append(elast2017+diff(xD2017(p2017), p2017)*p2017/xD2017(p2017))
EQNS.append(elast2018+diff(xD2018(p2018), p2018)*p2018/xD2018(p2018))
EQNS.append(elast2019+diff(xD2019(p2019), p2019)*p2019/xD2019(p2019))
EQNS.append(elast2020+diff(xD2020(p2020), p2020)*p2020/xD2020(p2020))
EQNS.append(elast2022+diff(xD2022(p2022), p2022)*p2022/xD2022(p2022))
EQNS.append(elast2023+diff(xD2023(p2023), p2023)*p2023/xD2023(p2023))
#### Market clearing conditions
EQNS.append(xD2016(p2016) - H2016)
EQNS.append(xD2017(p2017) - H2017)
EQNS.append(xD2018(p2018) - H2018)
EQNS.append(xD2019(p2019) - H2019)
EQNS.append(xD2020(p2020) - H2020)
EQNS.append(xD2022(p2022) - H2022)
```

```
EQNS.append(xD2023(p2023) - H2023)
#### Assumption on demand shock
student_pop_ratio_16_23 = StNH2016 / StNH2023
student_pop_ratio_17_23 = StNH2017 / StNH2023
student_pop_ratio_18_23 = StNH2018 / StNH2023
student_pop_ratio_19_23 = StNH2019 / StNH2023
student_pop_ratio_20_23 = StNH2020 / StNH2023
student_pop_ratio_22_23 = StNH2022 / StNH2023
EQNS.append(xD2016(p2023) - student_pop_ratio_16_23 * xD2023(p2023))
EQNS.append(xD2017(p2023) - student_pop_ratio_17_23 * xD2023(p2023))
EQNS.append(xD2018(p2023) - student_pop_ratio_18_23 * xD2023(p2023))
EQNS.append(xD2019(p2023) - student_pop_ratio_19_23 * xD2023(p2023))
EQNS.append(xD2020(p2023) - student_pop_ratio_20_23 * xD2023(p2023))
EQNS.append(xD2022(p2023) - student_pop_ratio_22_23 * xD2023(p2023))

UNKNS = [p2016,p2017,p2018,p2019,p2020,p2022,p2023,\
        A2016,B2016,A2017,B2017,\
        A2018,B2018,A2020,B2020,\
        A2019,B2019,A2023,B2023,\
        A2022,B2022,H2016,H2017,\
        H2018,H2019,H2020,H2022,H2023,\
        StNH2016,StNH2017,StNH2018,StNH2019,StNH2020,StNH2022,StNH2023,\
        elast2016,elast2017,elast2018,elast2019,elast2020,elast2022,elast2023]

#remove the values we'd like to solve for
EQNS2 = EQNS.copy()
EQNS2.pop(14)
EQNS2.pop(14)
EQNS2.pop(14)
EQNS2.pop(14)
EQNS2.pop(14)
EQNS2.pop(14)
print(EQNS2)


## minimising distance routine
## initialising distance at an arbitrary high number
current_mindistance=10**9

# Elasticity values to consider
elasticity_values = np.linspace(0.2, 3, 30)

# List to store distances between the modelled and real prices, for each
elasticity value
distances = []

# Iterate over each elasticity value and compute the distance
for elasticity in elasticity_values:
```

```
    # Update elasticity values in EQNS2 with the current elasticity
   EQNS_updated = [eqn.subs({elasticity_in: elasticity}) for eqn in EQNS2]

    # Solve the equations with updated elasticity
   ANS = solve(EQNS_updated, UNKNS, dict=True)[0]

    # List the modelled prices for this elasticity
   modelled_prices = [ANS[p2016], ANS[p2017], ANS[p2018], ANS[p2019],
ANS[p2020], ANS[p2022], ANS[p2023]]

    # Calculate distance between the modelled and real price
   distance = np.linalg.norm(np.array(modelled_prices, dtype=np.float64) -
np.array([
       Monthlyrent2016, Monthlyrent2017, Monthlyrent2018, Monthlyrent2019,
Monthlyrent2020, Monthlyrent2022, Monthlyrent2023], dtype=np.float64))
    ### numpy doesn't know how to handle sympy's Float type.
   distances.append(distance)
   if distance<current_mindistance:       ### if we have found a new candidate
argmin, we store it
       current_mindistance=distance.copy()
       current_argmin=ANS.copy()
       best_elasticity=elasticity.copy()

# Find the index of the minimum distance
min_distance_index = np.argmin(distances)

# Best elasticity corresponding to the minimum distance
best_elasticity = elasticity_values[min_distance_index]

print("Best elasticity:", best_elasticity)
print("Retained parametrisation:", current_argmin)


### Perform counterfactuals here
#defining more symbols
H2024,StNH2024,elast2024,A2024,B2024,p2024 = symbols("\
H2024 StNH2024 elast2024 A2024 B2024 p2024")

def xD2024(p):
    return A2024-B2024*p
EQNS2024=[]
EQNS2024.append(H2024 - 2718)   ### Manipulate supply as desired
EQNS2024.append(StNH2024 - 6000)  ### Manipulate demand as desired
EQNS2024.append(elast2024 - best_elasticity) ### Use calibrated elasticity
EQNS2024.append(elast2024+diff(xD2024(p2024), p2024)*p2024/xD2024(p2024))
student_pop_ratio_24_23 = StNH2024 / StNH2023
EQNS2024.append(xD2024(p2023) - student_pop_ratio_24_23 * xD2023(p2023))
```

```
EQNS2024.append(xD2024(p2024) - H2024)

UNKNS2024 = [H2024,StNH2024,StNH2022,elast2024,A2024,B2024,p2024,p2023]

EQNS2024b=[eqn.subs({p2023: current_argmin[p2023], A2023:
current_argmin[A2023], B2023: current_argmin[B2023], StNH2023:
current_argmin[StNH2023]}) for eqn in EQNS2024]

solve(EQNS2024b, UNKNS2024, dict=True)[0]
```