Department of Economics

STUDENT & UNIT DETAILS – TO BE COMPLETED BY THE STUDENT

Candidate Number	03907
Unit Name and Code	Practice track ES50156

DECLARATION – PLEASE READ CAREFULLY

When you enrolled as a student at the University of Bath, you agreed to abide by the University's rules and regulations and agreed that you would access and read your programme handbook. This handbook contains references to, and penalties for, unfair practices such as collusion, plagiarism, fabrication or falsification. The University's Quality Assurance Code of Practice, QA53 Examination and Assessment Offences, sets out the consequences of committing an offence and the penalties that might be applied.

By submitting this assessment, you confirm that:

1. You have not impersonated, or allowed yourself to be impersonated by, any person for the purposes of this assessment.
2. This assessment is your original work and no part of it has been copied from any other source except where due acknowledgement is made.
3. You have not previously submitted this work for any other unit/course.
4. You give permission for your assessment response to be reproduced, communicated, compared and archived for plagiarism detection, benchmarking or educational purposes.
5. You understand that plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to disciplinary action.
6. No part of this assessment has been produced for, or communicated to, you by any other person.

MARK AND COMMENTS – TO BE COMPLETED BY THE MARKER

	MARK (%)

Department of Economics

STUDENT & UNIT DETAILS – TO BE COMPLETED BY THE STUDENT

Candidate Number	03906
Unit Name and Code	Practice track ES50156

DECLARATION – PLEASE READ CAREFULLY

When you enrolled as a student at the University of Bath, you agreed to abide by the University's rules and regulations and agreed that you would access and read your programme handbook. This handbook contains references to, and penalties for, unfair practices such as collusion, plagiarism, fabrication or falsification. The University's Quality Assurance Code of Practice, QA53 Examination and Assessment Offences, sets out the consequences of committing an offence and the penalties that might be applied.

By submitting this assessment, you confirm that:

1. You have not impersonated, or allowed yourself to be impersonated by, any person for the purposes of this assessment.
2. This assessment is your original work and no part of it has been copied from any other source except where due acknowledgement is made.
3. You have not previously submitted this work for any other unit/course.
4. You give permission for your assessment response to be reproduced, communicated, compared and archived for plagiarism detection, benchmarking or educational purposes.
5. You understand that plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to disciplinary action.
6. No part of this assessment has been produced for, or communicated to, you by any other person.

MARK AND COMMENTS – TO BE COMPLETED BY THE MARKER

	MARK (%)

TWO NEW HOUSING MARKET INDICATORS:

THE AUTOREGRESSIVE MIXED EFFECTS HOUSE PRICE MODEL AND THE AUCTION DISCOUNT INDEX

DISSERTATION
submitted in partial fulfilment
of the requirements for the degree of
Economics for Business Intelligence and Systems MSc

by

Ali Rashid and Eloise Morrison-Clare

Department of Economics
University of Bath

16th October 2023

ABSTRACT

Accurate house price indices are essential for guiding macroeconomic policy. Fathom Consulting use Gao and Wang's (2007) Unbalanced Panel model for their own house price indices. The model however is oversimplistic, mis specifying the error structure and suffering from sample selection bias. We form our own variant of Nagaraja et al. (2011) model, coined the Autoregressive Mixed Effects Model. The model corrects the inefficiencies of the Unbalanced Panel method and is less prone to selection bias. The two models, UP and ARME, were trained and tested on the HM Land Registry Price Paid dataset. The ARME model outperformed the UP model in root mean squared error. Our results also suggest that the ARME model is more representative of the overall UK housing market. Using the obtained ARME model and auction data supplied by EIG, we constructed the Auction Discount Index (ADI), an index tracking the median discount (auction price divided by our estimate of its conventional market price) for properties sold at auction in each month. Using Toda and Yamamoto's (1995) Augmented Granger-Causality Test, we find evidence to suggest the ADI can help predict future housing market trends.

STATEMENT OF CONTRIBUTIONS

The work presented in this report would not have been possible without the contribution of our project supervisor Andrew Brigden, to whom we are very grateful. We would also like to thank Dr Chaitra Nagaraja who provided us with some auxiliary code from her 2011 paper, An Autoregressive Approach To House Price Modelling (Chaitra H. Nagaraja, 2011)

Due to the collaborative nature of this project, we highlight the sections of the report that have been written by each author:

Sections 1, 2, 3.1, 5: Eloise Morrison-Clare

Section 3.2, 3.3, 4, 6, 7, 8, Appendix A, B, C, D: Ali Rashid

CONTENTS

1 **Introduction** 1

2 **Literature Review** 2

2.1 Estimation Methods 2
2.2 The Unbalanced Panel model 4
2.3 The Autoregressive model 4

3 **Data and Methodology** 6

3.1 House price methods 6
3.1.1 The Unbalanced Panel Method 6
3.1.1.1 Model Description 6
3.1.1.2 Model Implementation 8
3.1.2 The Autoregressive Model 9
3.1.2.1 Model Description 9
3.1.2.2 Model Implementation 12
3.2 Land Registry Data 14
3.3 Model Testing and Performance Criteria 17

4 **Results and Discussion** 20

4.1 Estimation Results 20
4.2 Model Performance 22
4.3 Model Diagnostics 24
4.4 Discussion 30

5 **Theoretical Background** 32

6 **Data and Methodology** 33

6.1 Auction Sales Data 33
6.2 Toda and Yamamoto augmented Granger causality framework 34

7 **Results and Discussion** **37**

7.1 The auction discount index over time 37
7.2 Granger-causality 38

8 **Conclusions** **43**

9 **References** **45**

10 **Appendices** **48**

10.1 Appendix A 48
10.2 Appendix B 58
10.3 Appendix C 59
10.4 Appendix D 64

ILLUSTRATIONS

Figure 1:Land Registry Price Paid Dataset Property Sales Frequency Distribution 15

Figure 2: Visual breakdown of UK Postcode nomenclature (Allies Computing, n.d.) 15

Figure 3: Comparison between four house price indices 24

Figure 4: Serial Correlation for Sector SK10 2 26

Figure 5: Residual Distributions 27

Figure 6: Random Effects Q-Q Plot 29

Figure 7: Auction Discount Over Time 38

TABLES

Table 1: Property transactions divided by postcode levels and time frequencies16

Table 2: ARME Parameter Estimates 20

Table 3: Performance results of the three models 23

Table 4: ADF tests 39

Table 5: Information Criteria at different lag lengths 40

Table 6: Lagrange Multiplier test for Autocorrelation results 41

Table 7: Granger causality test 42

1 INTRODUCTION

Creating a reliable housing index is no easy feat, this is why multitudes of indices are proposed by the public and private sector. In the UK the Governmental house price indices are constructed from vastly different models which are vastly different to private sector models and so on. Fathom Consulting developed a Housing Index using an Unbalanced Panel model which has achieved some success compared to another private index by Zoopla Limited. Fathoms' model produced less biased results but produced larger standard errors than Zoopla's index. This is impressive considering the fact that the data Fathom used was public UK Land Registry data, compared to Zoopla who have detailed hedonic data available to them. Initially, we planned to use the Unbalanced Panel model to construct an Auction Discount Index. The use of this Discount Index is to compare the value of auctioned property to conventional property, that is houses sold via estate agents. This is a novel idea that theorises that the discount of auctioned property compared to conventional property could have some predictive power of the movements of the housing market as a whole.

The original Unbalanced Panel model however suffers from many faults. Consequently, we create our own house price model, the Autoregressive Mixed Effects (ARME) model, to remedy some of these issues. The ARME model is based upon the model found in Nagaraja et al. (2011), using similar model assumptions with a different estimation procedure. We extend our research to compare the Unbalanced Panel Model against this Autoregressive model. More accurate house price valuations will lead to, not only a more efficient index, but improvements on the accuracy of the Auction Discount Index as well. This paper is structured as followed.

Part one contains a comprehensive literature review that will cover the history of house price modelling. Then a methodology section detailing the models used in this paper. This is followed by results and discussion.

Part two begins with a background of the Auction Discount Index, then details of hypothesis testing and their results. Finally the paper will conclude with an overview of the findings.

2 Literature Review

2.1 Estimation Methods

Several methods have been used to create a reliable House Price Index, the most simplistic approach uses summary statistics, such as the mean and median. This method is flawed due to its reductionist nature. For example, the index does not include any adjustment for the quality of houses sold. There may be a change in the observed price in houses over time, but the model fails to distinguish whether this is due to an actual change in house price level or the chance that in different periods, a different mix (quality) of houses have been sold. A second group of methods utilise a hedonic regression. These models predict house prices from their observed characteristics and a set of fixed time effects. The coefficients on the characteristic variables act as shadow prices, they represent the change in house price for a marginal change in characteristic. The time fixed effects are used to model the trend in sales prices over time. These models are popular, for instance in the UK, the HM land registry uses a hedonic model (HM Land Registry, 2022). However, their disadvantage lies with correctly specifying the explanatory variables and collecting accurate data to represent relevant characteristics (Arthur Grimes, 2010).

A third set of methods are repeat sales which utilise homes that sell twice or more in their index. This method was initially proposed by Bailey et al (Bailey, 1963) and has

been extended by Case and Shiller (CS) (Case K. E., 1987) to incorporate heteroskedastic error terms rather than using a constant error term. The Case-Shiller model is so renowned that it is used to calculate S&P Home Price Indices in the US (S&P Dow Jones Indices, 2023). Both models construct an index by linear regression using the log price difference of two successive sales of a property. The earlier sale acts as a proxy for hedonic information since the previous price gives some indication of the home's quality. This is an advantage of Repeat sales methods because instead of carefully choosing explanatory variables, the hedonic information can be captured by the previous sale price. This was highlighted by Case and Quigley (Case B. Q., 1991) who proposed a hybrid model combining repeat sales and hedonic methodology and remarked that collecting hedonic data on a broad scale was too complex.

Repeat sales models trivialise a major characteristic of the housing market which is that houses are heterogenous goods. They assume that the quality of a house remains the same over time. In reality houses are maintained to different degrees. For instance their hedonic characteristics could improve through renovations and extensions or diminish through neglect. Thus, simply capturing this information in previous sale price could lead to poor estimations, especially if the previous sale occurred a long time ago. Some literature adjusts for quality changes, for example Goetzmann and Spiegel (Goetzmann W. N., 1995) correct for changes in quality according to when a property was sold. Another paper, written by Case and Quigley (Case B. Q., 1991) suggests that houses naturally depreciate in quality with age. Palmquist (Palmquist, 1982) computes an independent depreciation factor to capture the impact of age on a home's price level.

A second limitation of repeat sales techniques is that they may be unrepresentative of the housing market as a whole. They utilise properties which have been sold twice or more so they may inaccurately estimate the value of homes which have sold only once. Englund et al. (Englund, 1999) and Meese et al. (Meese, 1997) both find that there is a difference between repeat and single sale homes. They found repeat sale

homes were older, smaller, and more "modest" than single sale homes. These differences suggest that by only including a subset of all sales, repeat sales indices suffer from sample selection bias.

Despite their disadvantages, repeat sales are valuable in their simplicity and reduced data requirements. The models used in this paper have been chosen to exploit these features. These are the Unbalanced Panel (UP) first proposed by Gao and Wang (Gao, 2007) and The Autoregressive Mixed Effects model (ARME) adapted from Nagaraja et al. (Chaitra H. Nagaraja, 2011). The models differ in construction but have a variety of advantages over previous housing indices, these will be discussed in the following sections and the methodology.

2.2 THE UNBALANCED PANEL MODEL

The UP model exploits the simplicity of repeat sales models in the sense that a house's hedonic information is contained within previous sale prices. In literature, there is mixed evidence to support the use of this model to construct a housing index. Grimes et al. (Grimes, 2021) compare the residual mean squared error (RMSE) of house price indices generated from the CS and UP methods. They find that that the UP model produces better estimates than the CS model. This is very encouraging as the CS model is popularly compared in comparison studies as a benchmark index. However, in the same paper it is found that when the error generating process varies, so does the most reliable index. The CS model is a more efficient estimator if the errors follow a random walk process, but the UP model is better if they are stationary. There is a large body of research that concurs that the underlying data generation process of house prices will affect the efficiency of an index (Abraham, 1991) (Goetzmann W. N., 1992) (Wang, 1997). Therefore, the positive results of the UP model may vary.

2.3 THE AUTOREGRESSIVE MODEL

The Autoregressive Index proposed by Nagaraja et al. (Chaitra H. Nagaraja, 2011) is shown to be a better predictor for house prices than the CS model. The results show the RMSE for their ARME index is lower than the CS index for 19 of 20 US metropolitan areas. The first advantage of this model is that the previous sale price becomes less important overtime as it enters through the AR (1) component, this feature will be discussed mathematically in the methodology section. This feature is more realistic than other repeat sales methods, as the hedonic information contained in the previous sale price becomes less relevant with a longer gap time between sales as the property is more likely to have undergone changes. The second advantage is that the model can be used to estimate all sales, single and repeat. This removes sample selection bias that has been a disadvantage of previous models.

3 DATA AND METHODOLOGY

3.1 HOUSE PRICE METHODS

3.1.1 The Unbalanced Panel Method

3.1.1.1 Model Description

The UP model was originally proposed by Gao and Wang (Gao, 2007) and incorporates all sales information on a property that has sold twice or more. The model utilises an OLS panel regression which can be written in matrix form as followed:

$$\ln \mathrm{HP}_{\mathrm{it}} = M_t\beta + A_i\tau + \varepsilon_{it} \ , \ t = 1, \dots, T, \ i = 1, \dots, N$$

(1)

$t = 1, \dots, T$ are the discrete time periods observed in the model and $i = 1, \dots, N$ are the house's identifiers so N is the total number of observations across the dataset. This model proposes that the current log worth of the house, $\ln \mathrm{HP}_{\mathrm{it}}$, is determined by two components. The first is a time trend, β, this is what the house was worth in 1996 (the first year of observation included in the data set) scaled by the 2023 beta coefficient. Essentially, this scales up the houses worth by a market index over the observed time periods. The second component is the houses intrinsic value τ, which is the parameter that estimates the house's value in the absence of the time trend. Both components are assumed to be fixed effects, this assumes the house's intrinsic

value to be constant over time, however this is not often the case in reality. As mentioned in the literature review section, houses vary with quality over time which will affect their value. Since this is not a consideration in this model this may lead to inefficiency in estimation. The noise term, ε_{it}, represents the deviation of the estimated log price from expected/actual log price. In this model the residual is assumed to be stationary and follows

$$E(\varepsilon^i) = 0\ , E(\varepsilon^i \varepsilon^{l\prime}) = \begin{cases} \sigma_\varepsilon^2 I_{ni \times ni}, i = l \\ 0, i \neq l \end{cases}$$

(2)

The UP differs from conventional panel data modelling because the individual houses are only observed/sold in a few time periods, not in all periods recorded in the data. The explanatory variables, the house's intrinsic value, τ, and the market index, β, are treated as fixed effects. This means that all of the parameters are non-random. Estimation of the model parameters becomes relatively simple by virtue of the fact that the model is linear, and they can be fit by Ordinary Least Squares (OLS). Furthermore, using fixed effects incorrectly will not bias the model (Hausman, 1978). The fixed effects both depend on the market index parameter β, which suggests that every house's intrinsic value is subject to fluctuations in this parameter. This makes sense as one could suggest that in periods of economic downturn the aggregate quality of house's depreciates and vice versa, but it may be too general to treat houses as such homogenous goods. The choice of random effects could be a better specification for individual house effects to capture their heterogeneity. This will be discussed further in section 3.1.2.1.

3.1.1.2 Model Implementation

The UP panel is implemented in python and initially loops through the postcode sectors to create the matrix $M_t = (M_1, \dots M_T)'$ which are the time fixed effect dummies. When a sale of house i has taken place in period t the cell contains 1 and if there was no sale it contains 0. The time trend component, β, is constructed by taking the average house price in absence of time trend, $\bar{A}$, and multiplying by the exponential market index value for every period and so, $\beta = (\bar{A}e^{\beta_1}, \dots, \bar{A}e^{\beta_T})'$. Similarly to M_t, the matrix $A_i = (A_1, \dots, A_N)'$ contains the individual house fixed effects dummies which multiplies by the parameter $\tau = (\tau_i, \dots, \tau_N)'$. The code is found in Appendix A.

There is a collinearity problem between the parameters β and τ, therefore they cannot be independently determined. Thus, the following restriction is imposed on τ. In words, this removes the first year of observation from the dataset.

$$\tau_1 = -\sum_{i=2}^{N} \tau_1 \qquad (3)$$

The model parameters are then estimated through an OLS linear regression. The UP results will be discussed and compared to the ARME model in section 4.

3.1.2 The Autoregressive Model

3.1.2.1 Model Description

The ARME model was adapted from Nagaraja et al. (Chaitra H. Nagaraja, 2011). In their model each house is identified by a unique ID, i, and a sale ID, j, which is the same in our adaptation. However, houses are identified by an additional index, z, which locates each observation by US metropolitan area. The z index is used to model a ZIP code random effect. This means that the houses used in estimation are subject to a random independent variable which, in this case, is a parameter τ which has its own statistical distribution for each ZIP code, z. In this paper we include a random effect for each individual house rather than grouping them by z. Qilong remarks that in Unbalanced panel studies, the individuals may differ among themselves rather than as a whole group, and so random effects is the preferred specification (Qilong, 2020). Each house will vary and correlate with itself, not the whole population. This captures the heterogeneity of the housing market which could make the ARME model more realistic.

The model provides two regression equations which estimate for single and repeat sales. As discussed in the literature review, this is an important property of a housing index as it better represents the entire housing market. Single sale properties have been found to differ in characteristics from repeat sales and so having two estimation models improves its application to estimating the value of all properties. The regressions are

$$y_{i,1} = \mu + \beta_{t(i,1)} + \tau_i + \epsilon_{i,1}, \quad j = 1$$

$$y_{i,j} = \mu + \beta_{t_{(i,j)}} + \tau_i + \phi^{\gamma(i,j)} \left(y_{i,j_{-1}} - \mu - \beta_{t(i,j-1)} - \tau_i \right) + \epsilon_{i,j}, j > 1$$

(4)

$y_{i,j}$ measures log house price of house i and sale number j. For example, y_{23} is the log price of the 2nd house on its 3rd sale. μ is the overall mean of house prices across all time periods, $\beta_{t_{(i,j)}}$ is the log price index at t when the jth sale of house i takes place, these are assumed to be fixed effects. τ_i are random house effects for each house $\tau_1, \dots, \tau_N$, where N is the total number of observations used to fit the model. Since $\tau_i \sim \mathcal{N}(0, \sigma_\tau^2)$, every price estimation contains a random term component that varies within certain bounds. This is to incorporate the idea that properties will vary in quality. The error terms, $\epsilon_{i,j}$ are independent and are distributed as

$$\epsilon_{i_1} \sim \mathcal{N}\left(0, \frac{\sigma_\epsilon^2}{1-\phi^2}\right), \qquad \epsilon_{i_j} \sim \mathcal{N}\left(0, \frac{\sigma_\epsilon^2(1-\phi^{2\gamma(i,j)}}{1-\phi^2}\right)$$

(5)

When there is only one sale ($j = 1$) the error variance is the marginal variance. The autoregressive coefficient ϕ, $|\phi| < 1$ diminishes by the factor γ where $\gamma(i,j) = t(i,j) - t(i,j-1)$ which is the gap time between sales, so as the gap time increases, the previous sale becomes less useful in predicting the current price estimate. The variance of the errors , ϵ_{ij}, grows with the gap time, γ. This is logical seen as with a larger time period between sales it becomes more likely that the house has changed. This leads to larger errors in estimation.

The full autoregressive model is fit by

$$Ty = TX\mathrm{B} + TZ\tau + \epsilon^*$$

(6)

y is the vector of log prices, X and Z are the design matrices for the fixed effects $\mathrm{B} = [\mu, \beta_1 \dots \beta_T]'$ and the random effects respectively. These contain 0s and 1s to match the index of each element of y. The entire system is pre-multiplied by a Transformation matrix T, which is an $N \times N$ matrix that adds the autoregressive component to the model where $t_{(i,j)(i',j')}$ corresponds to the cell with (i,j) row and (i',j') column so T is constructed as

$$t_{(i,j)(i',j')} = \begin{cases} 1, & if\ i = i', j = j' \\ -\phi^{\gamma(i,j)}, & if\ i = i', j = j' + 1 \\ 0, & otherwise \end{cases}$$

(7)

In words, T is 1 when the cell corresponds to the most recent sale, T is $-\phi^{\gamma(i,j)}$ when the cell corresponds to the previous sale and 0 if the cell contains no sale of the *ith* house. The covariance matrix for the error term, $T\epsilon^* \sim \mathcal{N}\left(0, \frac{\sigma^2_\epsilon}{1-\phi^2} diag(r)\right)$ allows ϵ_{i_1} and ϵ_{i_j} to be combined. The matrix $diag(r)$ contains the diagonal elements given by

$$r_{i_j} = \begin{cases} 1 & , \quad j = 1 \\ 1 - \phi^{2\gamma(i,j)}, & j > 1 \end{cases}$$

(8)

The covariance matrix V of the error terms, $\epsilon = T\epsilon^*$, is split into the sum of variance contributions from the time series and random effects given as

$$V = \frac{\sigma_\epsilon^2}{1-\phi^2} diag(r) + (TZ)D(TZ)'$$

(9)

where $D = \sigma_\tau^2 I_Z$ and I_Z is an identity matrix of dimension $Z \times Z$

3.1.2.2 Model Implementation

The ARME model has the following error structure

$$y = X\mathrm{B} + Z\tau + u_j \ \rightarrow u_j = \begin{cases} \epsilon_j \ , j = 1 \\ \phi^\gamma u_{j-1} + \epsilon \ , j > 1 \end{cases}$$

(10)

When there is a single sale, the errors are not serially correlated, but they are heteroskedastic as the variance grows with the gap time γ. Equation *(4)* can be rearranged and factorised as follows

$$y_j - \ \phi^\gamma y_{j-1} = \mu(1-\phi^\gamma) + \tau_i(1-\phi^\gamma) + \beta_j - \phi^\gamma \beta_{j-1} + \epsilon$$

(11)

The only unknown parameters in this regression are ϕ and σ_ϵ^2 , they are estimated by a Prais-Winsten procedure, then the full model can be fit. The steps involved are outlined by the following algorithm.

ARME Model Fitting Algorithm

1. Run a linear mixed effects model for equation *(10)* to estimate $\widehat{u_j}$ and $\widehat{\sigma_\epsilon^2}$

2. Estimate $\boldsymbol{\phi}$ using Non-linear Least squares estimation, this is only for repeat sales and uses the equation $\boldsymbol{\phi}^{\gamma} u_{j-1} + \epsilon$

3. Using the estimates found in 1. construct the T and V matrices. Now $\hat{\mathrm{B}}$ and $\hat{\tau}$ are estimated through GLS using equation *(12)*.

4. Using $\hat{\mathrm{B}}$ and $\hat{\tau}$ repeat step 1 to estimate $\widehat{u_j}$ and $\widehat{\sigma_\epsilon^2}$

5. Repeat steps 2-4 until the estimates $\hat{\boldsymbol{\phi}}$ and $\widehat{\sigma_\epsilon^2}$ converge

6. Using $\hat{\boldsymbol{\phi}}$ and $\hat{V}$, repeat step 3 to get the final coefficient estimates $\hat{\mathrm{B}}$, $\hat{\tau}$, $\widehat{\sigma_\epsilon^2}$

In the interest of saving time, the estimates in this paper were calculated after one iteration. For more accurate estimates let $\hat{\boldsymbol{\phi}}$ and $\widehat{\boldsymbol{\sigma_\epsilon^2}}$ converge within a specified threshold. Since the errors are heteroskedastic, equation *(11)* is pre-multiplied by $P = V^{-\frac{1}{2}}$ to perform the Generalised Least Squares estimation (Appendix B).

$$PTy = XPT\mathrm{B} + PZ\tau + Pu_j$$

(12)

3.2 LAND REGISTRY DATA

The publicly available UK Land Registry dataset covers all single residential property sales in England and Wales from January 1995 to August 2023. The 30,052,312 rows detail the price, date, and the address of the sold property. Figure 1 displays the frequency distribution of properties based on the number of times each property is featured in the dataset. Evidently, most homes are only sold once. Repeat-sales model exclude these sales from the estimation process, resulting in the selection bias mentioned in section 2.1.

UK postcodes are made up of the outward code and the inward code, separated by a space. The outward code includes the postcode area followed by the postcode district. Similarly, the inward code includes the postcode sector and postcode unit. Figure 1 provides further clarity on the UK postcode format. Table 1 divides all unique postcodes listed in the dataset into these hierarchical categories, including the average number of property sales at each level of precision and data frequency. Lower location precision estimations benefit from a larger sample size producing more stable indices. However, it come at the cost of dissolving location specific hedonic information and adding computational complexity.

Registry, Office for National Statistics (ONS) , Land and Property Services Northern Ireland (LPSNI) and Registers of Scotland. As such, it would be interesting to compare the highly specified index to those estimated for the more parsimonious models.

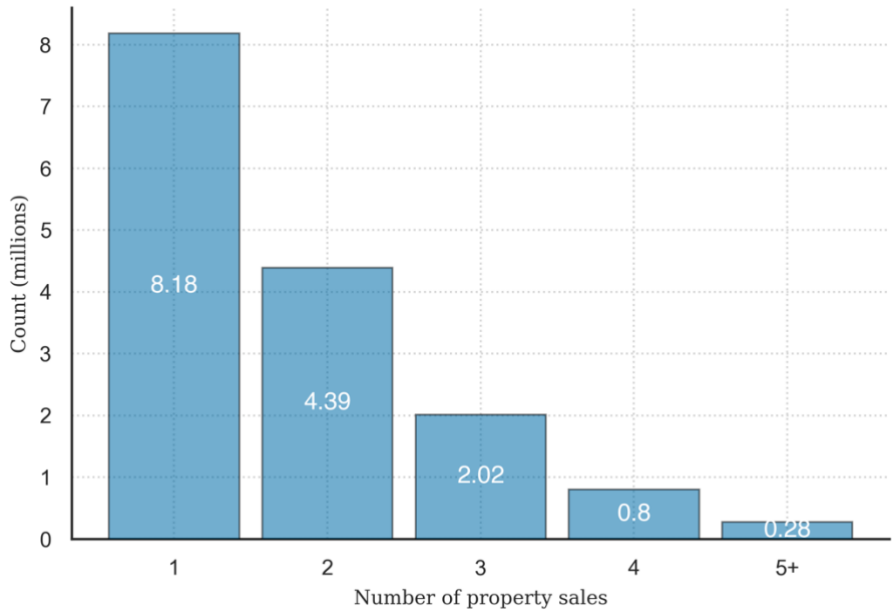

Figure 1:Land Registry Price Paid Dataset Property Sales Frequency Distribution

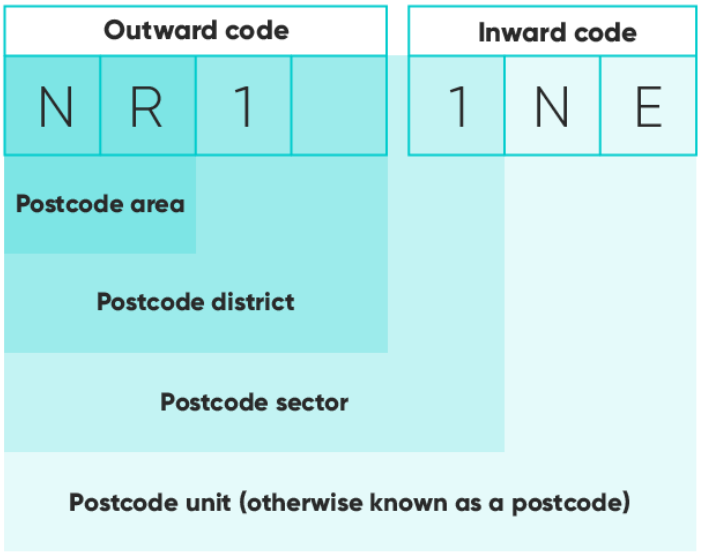

Figure 2: Visual breakdown of UK Postcode nomenclature (Allies Computing, n.d.)

The added value of local amenities, such as railway stations, exponentially decays per unit distance (Boham, 2016). Consequently, indices estimated from larger areas are prone to inaccuracy resulting from the drastic location heterogeneity of the homes within the sample. Additionally, large matrices are extremely resource intensive for coefficient estimation. The Unbalanced Panel model uses a Panel Least Squares estimation method with a Big O run-time: $O(K^2 * (N + K))$ per regression (Agus, 2015), with K the number of regressors and N the sample size. The regressors are the house effects, A which vary in size and the 28-year time effects, M. With this notation, the Big O runtime per iteration is: $O((A + M)^2 * (N + A + M))$. Comparing the average-case algorithm complexities of computing indices at the district and sector levels, the district estimation performs $\frac{2388}{9270} \approx \frac{1}{4}$ the number of iterations the sector algorithm would. However, each iteration would have around 4x the sample size N and properties A, increasing time complexity by approximately 64 times per loop relative to the equivalent algorithm partitioning sales by postcode sectors.

Table 1: Property transactions divided by postcode levels and time frequencies

	Postcode Levels			
	Area	**District**	**Sector**	**Unit**
Count	123	2,388	9,720	1,289,741
Average no. sales per postcode level	244,327	12,584	3,091	23
Average no. sales per frequency				
Year	8,725.99	449.45	110.42	0.83
Quarter	2,181.50	112.36	27.61	0.21
Month	727.17	37.45	9.20	0.07

Choosing the sample frequency rate requires similar considerations. Time fixed effect dummies compute the house price indices in both models, hence increasing the number of time periods, M will similarly increase algorithmic complexity

exponentially. Furthermore, more frequent data reduces the number of sales used to estimate each time effect.

Given the above, both models operate at the annual sector level, estimating 28 price indices for every postcode sector meeting the minimum sample size requirement. Cubic interpolation is then used to convert the estimates to a monthly series to smooth the transition from one period to the next.

3.3 Model Testing and Performance Criteria

With no "true" house price index to compare our models against, we follow the conventions proposed by the literature and use predictive ability as an objective measure of performance (Chaitra H. Nagaraja, 2011), (Grimes, 2021), (Jiang, 2015). The data is split into a training set for parameter estimation and a testing set for evaluation. Nagaraja, Brown and Zhao (2011) target an approximate 85:15 training to testing data split. They place the final sale for all homes that have sold at least three times into the test set. Then, for homes that have sold twice, their second sale enters the test set with a probability of 50%. We do the same except the probability is adjusted to 27% to meet the targeted 85% to 15% split for our sample (code in Appendix C).

During testing, the house prices of the previous sales of those in the test set are inflated (or deflated) using the price indices and any other necessary parameters estimated from the training set to estimate the worth of the house in the test set. The prediction error $(Y - \hat{Y})$ is then extracted for the standard performance metrics: mean error (ME), mean absolute error (MAE) and root mean squared error (RMSE). Their respective percentage counterparts are used for evaluation. ME tracks estimator bias, MAE measures the unweighted average magnitude of prediction error by the model and RMSE calculates the variance of the errors, hence measuring the efficiency of the model.

$$ME = \frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \widehat{Y}_i\right)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|Y_i - \widehat{Y}_i\right|$$

$$RMSE = \frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \widehat{Y}_i\right)^2$$

(13)

Measurement errors and immense outliers in the Land Registry dataset can only be identified and removed after the final sales have been tested. We generously remove 5% of the largest *absolute percentage errors* prior to calculating the results. It is important to note how the compositions of the two subsets affect the performance of the ARME and UP models. Properties that have only sold once are dropped prior to running the UP model since it is a repeat sales estimation method. Also note that the testing set by nature can only include sales from properties with multiple sales over the sample period. The UP indices are trained and tested purely on repeat sales. Contrast this with the ARME model which uses all sales in its estimation process. Even with the ARME model estimating from a larger sample size and downweighing single sales in its estimation process, it is fair to presume an advantage towards the UP model. Therefore, the ARME model is additionally reimplemented using only repeat sales for comparison, denoted as ARME – RS.

Additionally, a single weighted index is created to represent the population-wide house price inflation proposed by each model. In each period, the estimated time effect for each sector is multiplied by the proportion of transactions that occurred in the sector and then summed across to find the weighted time effect for the period.

The process is done for each period to form the weighted index, with the first observation normalised to 1. The three times series are graphically compared to the official UK House Price Index (HM Land Registry, 2022), a hedonic regression model. Although no "true" house price index exists, hedonic methods generally outperform repeat sales methods (Clapham, 2004). The UK HPI incorporates a variety of property specific characteristics, such as square footage, number of bedrooms and area demographics (Office for National Statistics, 2023). The index is jointly produced by the HM Land Registry, Office for National Statistics (ONS) , Land and Property Services Northern Ireland (LPSNI) and Registers of Scotland. It is interesting to compare how this highly specified index against the more parsimonious models.

4 Results and Discussion

4.1 Estimation Results

The ARME estimates for μ, representing the log average price for the sector, ϕ, the AR(1) coefficient, $\phi^{\bar{\gamma}}$, where $\bar{\gamma} = 6.7$: the mean gap time between sales in the training set, and σ_ϵ^2, the residual variance, are summarised for 10 different postcode sectors in Table 2. The random effect slope estimates were essentially zero for across all sectors, hence their exclusion from the table. The first four rows belong to the sectors with the most transactions in the training set. The following two sectors, S10 9 and TS1 3, juxtapose each other in average house prices, with Kensington being the most expensive area in the UK, and Middlesbrough the least. The final four sectors were chosen at random for cross-comparison.

Table 2: ARME Parameter Estimates

Sector	Locality	N	$\hat{\mu}$	$\hat{\phi}$	$\hat{\phi}^{\bar{\gamma}}$	$\widehat{\sigma_\epsilon^2}$
LE10 0	Hinckley, Leicester	11770	10.6960	1.0000	1.0000	0.0087
RG22 4	Basingstoke	11202	11.1276	0.9999	0.9996	0.0062
LU2 7	Luton, Bedfordshire	10666	10.8797	1.0000	1.0000	0.0075
GL2 4	Gloucester	10644	10.8676	0.9761	0.8504	0.0008
SW10 9	Kensington, London	5338	12.0849	0.9861	0.9108	0.0043
TS1 3	Middlesbrough	2152	10.1121	0.9481	0.6998	0.0094
CB23 7	Cambridge	3181	11.3182	0.9908	0.9398	0.0007
BS29 6	Banwell, Bristol	1364	10.8471	0.9728	0.8314	0.0023
NP26 4	Caldicot, Newport	2815	10.9316	0.9591	0.7558	0.0031
SS16 5	Basildon, Essex	4165	10.7728	0.9630	0.7770	0.0027

The μ estimates follow the general intuition of higher values to more expensive areas such as Kensington and Cambridge relative to that of Middlesbrough or Hinckley. The autoregressive parameter is close to 1 for eight of the sectors, and even rounds to unity for LE10 0 and LU2 7. Serial correlation however is not measured by ϕ, but by ϕ^{γ}. Examining ϕ^{γ} for the average γ, 6.7 years offers more clarity on the observed data generating process. Seven out of the ten sectors have $\hat{\phi}^{\overline{\gamma}}$ estimates sufficiently below 1 to suggest model stationarity. Conversely, the autocorrelation parameter for the top three sectors by sample size approximate $\hat{\phi} = 1$, implying houses prices in these areas follow a random walk process.

The average house price increase between t and $t-1$ for a given postcode sector is estimated in monetary terms as $\hat{\mu}\left(\frac{\hat{\beta}_t}{\hat{\beta}_{t-1}}\right)$ by the ARME model. A shock $u_{i,t}$ to property i at time t would push the one period growth of the house price: $Y_{i,t+1} - Y_{i,t}$, away from $\hat{\mu}\left(\frac{\hat{\beta}_{t+1}}{\hat{\beta}_t}\right)$ in the following period due to $u_{i,t+1} = \phi^{\gamma} u_{i,t} + \epsilon$. With an autocorrelation coefficient $\phi < 1$, ϕ^{γ} decreases monotonically between each period, resulting in the growth rate of y_i returning to the average trend for the sector. Conversely, $\phi = 1$ suggests house price deviations permanently affect $Y_{i,t'}$ where $t' > t$.

Coefficient ϕ rounds to 1 (to 3 significant figures) for around 30% of sectors. Before concluding house prices in these sectors follow random walk processes, it should be noted that these sectors generally have the largest sample sizes, a trend present in Table 2. For computational ease, the iteration loop to estimate $\hat{\phi}$ and $\hat{\sigma}_{\epsilon}^2$ is only repeated once in our estimation process. As consequence, it is likely the obtained estimates of the autocorrelation parameter are suboptimal. This concern is strengthened for sectors with more data, as more iterations are likely needed for log-likelihood maximising coefficients. Despite these fears, the population estimates are still likely consistent and asymptotically unbiased (Section 4.3). Across all sectors, the

ARME model calculates the average $\hat{\phi} = 0.978$, and the average persistence between sale pairs, $(y_{i,j}, y_{i,j-1})$, : $\bar{\phi}^{\bar{y}} = 0.861$.

4.2 MODEL PERFORMANCE

Table 3 summarises the predictive performance for the UP, ARME and the ARME – RS models in both pound sterling and percentage terms. The UP and ARME models are the focus of comparison. Both models exhibit similar magnitudes of estimator bias according to their ME scores. The bias however occurs in opposing directions, with ARME exhibiting upward bias while the UP model is biased downwards. In percentage terms, the size of the bias is a marginal, just under 1%. The models however rank differently when looking at MAE and RMSE. MAE simply averages over the all the absolute errors in the testing set predictions. Therefore, the UP model, with its lower MAE score (£17802), on average provides smaller forecasting errors relative to the ARME model. RMSE however averages over the squared errors before the final square root operation. Consequently, RSME is more sensitive to larger prediction errors. So, although the UP model performs better than the ARME model on average prediction, the variance across the errors is larger, hence more prone to producing extreme forecasting errors.

If we also consider the ARME – RS model's results, it is clearly superior to the two other models by all three metrics. ARME – RS is identical to ARME in its estimation procedure, except it only uses repeat-sales observations. Therefore, its enhanced predictive capability relative to the ARME model evidences the differing characteristics between repeat-sale and singe sale homes. If both subsets were similar, the standard ARME model would perform better than its repeat-sales only varying due to the increased sample size. Therefore, methods trained and tested using only repeat-sales performance superiorly relative to models that also include other sales into the training process.

To remove this comparative advantage, ARME – RS is compared to the UP model. The ARME – RS model slightly beats the UP model in ME. The relative performance enhancement however is marginal, with both MAE scores rounding to 6.6%. The differing capabilities of the two models however is shown in the drastically lower RSME for the ARME – RS model. These findings show that ARME outperforms the UP when tested and trained on the same data.

Table 3: Performance results of the three models

	UP	**ARME**	**ARME - RS**
ME	3991 (0.97%)	-3147 (-0.20%)	-1228 (0.08%)
MAE	17802 (6.63%)	18006 (6.75%)	17602 (6.55%)
RMSE	68160 (11.9%)	54136 (11.1%)	53994 (11.1%)

Note: metric given in terms of pound sterling

The weighted time series from each model covering UK house price inflation for England and Wales are compared to each other and to the UK HPI (Figure 3). The UK HPI is constructed with extensive hedonic data over a larger time span and

provides more detailed short-run dynamic shifts than the other indices, especially in the 90s and early 2000s. The three other series are almost vertical in this period. The shared behaviour between the series is unsurprising seeing the similarity between the estimation methods. Up until 2016, the four indices track each other well, with no obvious lead/lag relationship between them. After 2016, the UP and ARME – RS indices diverge in opposing directions away from the UK HPI. ARME however trends closely to the UK HPI over most of the sample period. The similarity between the highly stylised, data intensive UK HPI and the parsimonious ARME model is

impressive. It emphasises the benefit adding single sales into the estimation procedure has on representing price trends for the whole housing market.

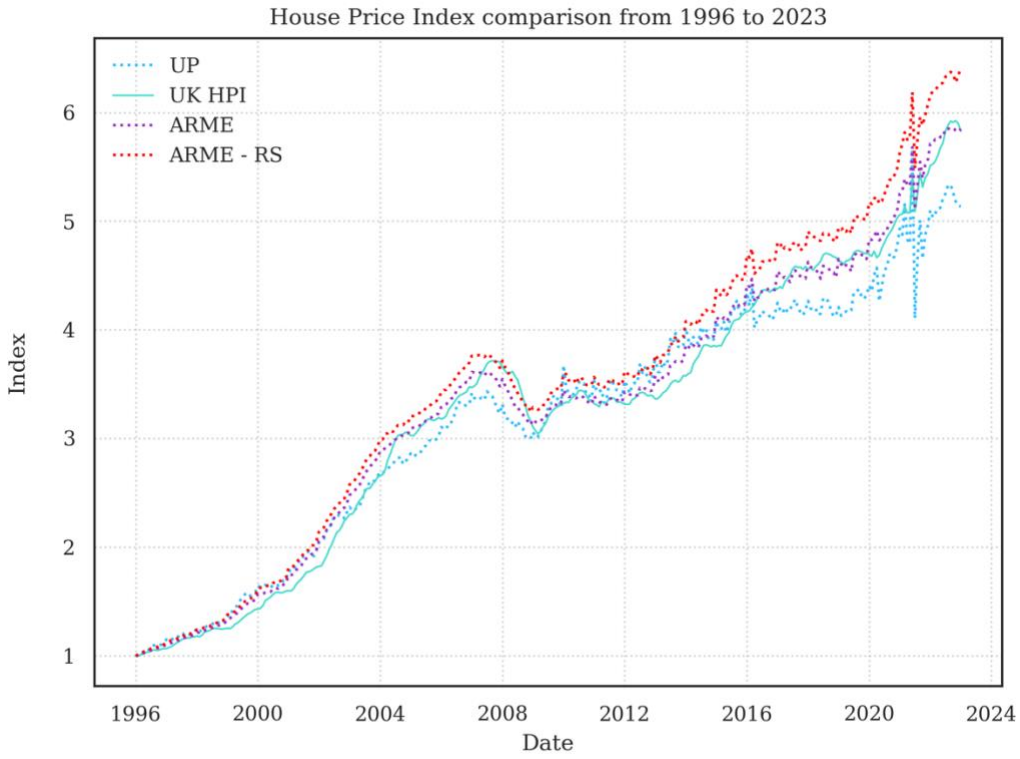

Figure 3: Comparison between four house price indices

4.3 MODEL DIAGNOSTICS

The following four assumptions were used while constructing the ARME model:

1. Residuals suffer from serial correlation, defined by $u_{i,j} = \phi^{\gamma(i,j)} u_{i,j-1} + \epsilon_{i,j}$
2. The residuals are normally distributed
3. The residuals are inertly heteroscedastic, and must be corrected for
4. Random effects are normally distributed of mean zero and constant variance: $\tau_i \sim N(0, \sigma_\tau^2)$

The assumptions are tested on the training dataset, with some being more crucial than others. For any repeat sale, the ARME model posits that its error is correlated to the error of its previous sale by the factor ϕ^{γ}. To check this, the residuals of the repeat sales are paired together with the residual of their previous sale post ARME estimation. The pairs are then sorted into groups based on the number of years between the two sales, γ. The correlations between the repeat sale residuals and the previous sale residuals, ϕ, are computed for each group. Finally, ϕ^{γ} is plotted against γ, with the hypothesised relationship also modelled for comparison. For added context, groups with fewer than 10 pairs were deemed to be potentially misleading and hence removed. Correlations computed with fewer than 25 residual pairs are coloured blue, and all else red. Sector SK10 2, with $\hat{\phi} \approx 0.96$, is graphed as an example (Figure 4), but the pattern is the norm for most of the sectors examined. As theorised, residual correlation is inversely related to the time between sales.

The remaining residual distribution assumptions are examined, that they are normally distributed, zero-meaned and heteroscedastic for UP estimates. The ARME model corrects for heteroscedasticity via its GLS estimation. Therefore, residuals estimated using the ARME model should have a constant variance.

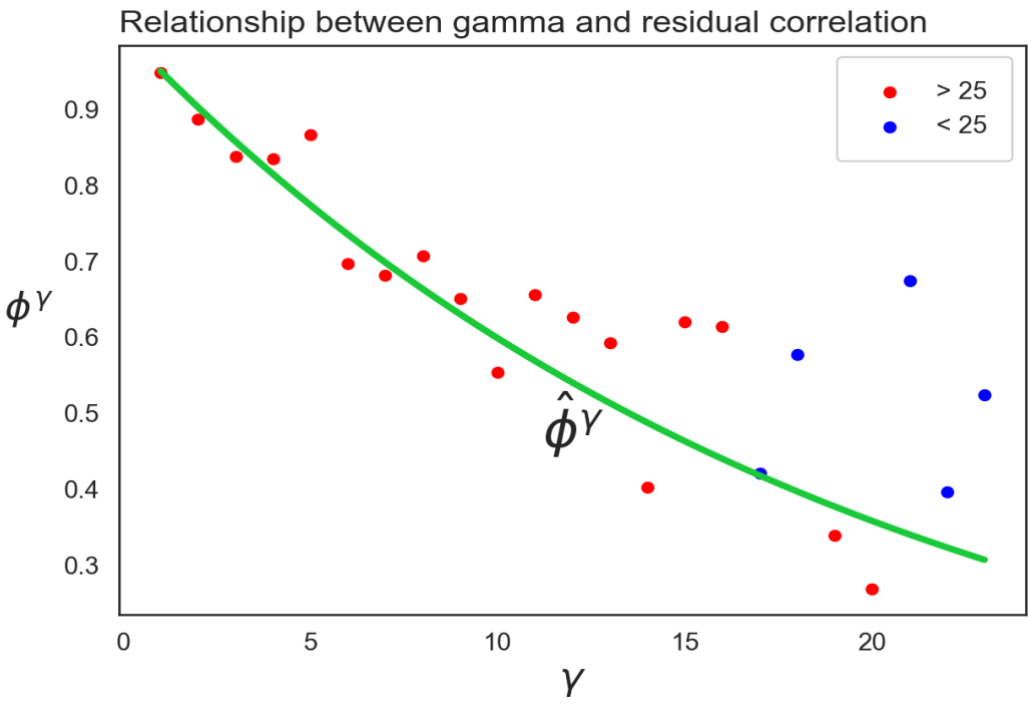

Figure 4: Serial Correlation for Sector SK10 2

Residual normality is investigated through a residual histogram and a Normal Q-Q plot. For the Q-Q plot, the residual series is sorted by size and split into H quantiles where H is the number of obtained residuals. Then, the hypothesised distribution $u \sim N\left(0, \frac{\sigma^2}{1-\phi^2}\right)$ is simulated and similarly partitioned into H quantiles. If the residuals are normally distributed, quantiles of the residual series should match that of the simulated probability distribution. Hence, plotting the quantiles of the observed sample against the theoretical should reveal an approximate 45° line if both distributions are similar.

The Breusch-Pagan (1979) tests for heteroscedasticity by running an auxiliary regression between the squares of the residual estimates on the model regressors. Heteroscedasticity arises from the non-independence between the error term and the independent variables of model. Therefore, if the coefficients of the auxiliary

regression are statistically different from zero, as tested via a chi squared statistic with K degrees of freedom, heteroscedasticity is concluded.

Residuals from 30 different, randomly chosen postcode sectors were estimated using both models, with the corresponding histograms and Q-Q plots in Figure 5. Both sets of estimates have similar leptokurtic distributions, exhibiting extreme kurtosis as evidenced from the backwards *S* shaped Q-Q plots. The extremity resulted in us performing a log transformation on the y-axis of the density plots for visual tractability, otherwise the plot would simply be large bar in the centre and not much else. The fatter tails and larger deviations of the ARME plots suggests a poorer handling of fitting outliers than the UP model, a hardly surprising result due to the higher sample heterogeneity used in the ARME estimations.

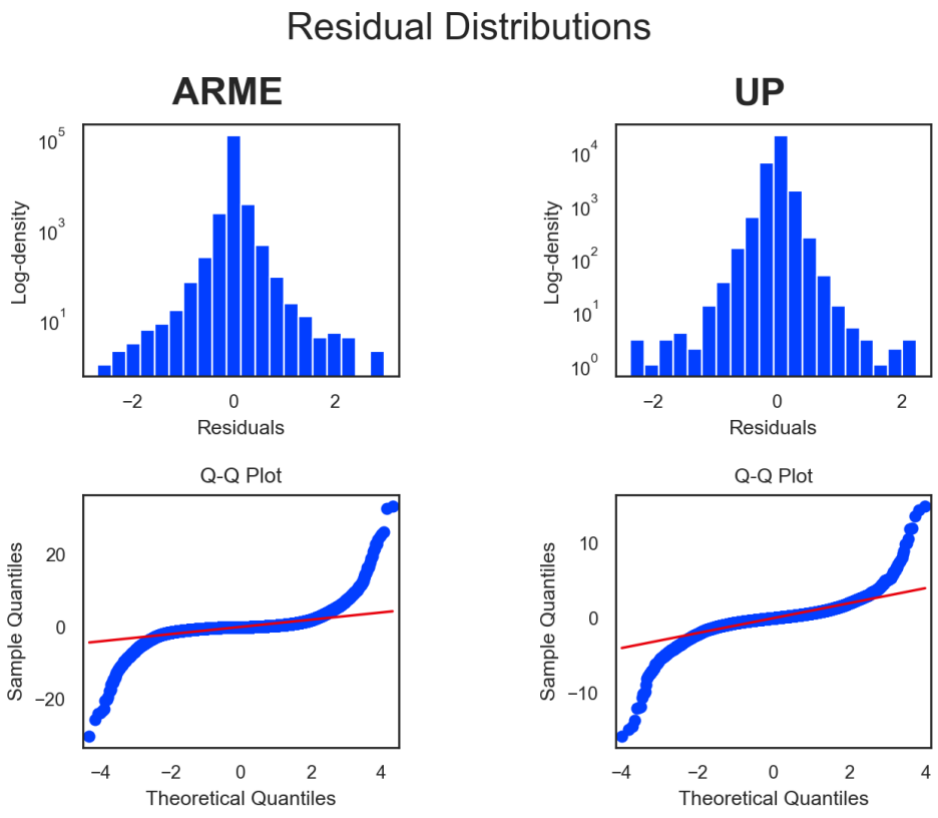

Figure 5: Residual Distributions

The Breusch-Pagan (Breusch, 1979) test was performed individually on the same 30 sectors. The test strongly rejected the null hypothesis of homoscedasticity at the 1% significance level for the residual series obtained from every estimation performed by the UP method. Contrarily, only 5 out of 30 postcode sectors had p-values below 0.05 for the ARME estimates (Appendix D). The tests imply that the GLS technique removed the heteroscedasticity for most, but not all house price indices.

The residual diagnostic tests affirm that the ARME model mostly adjusts for the serial correlation and heteroscedasticity in the residuals but also evidences their non-normality. The residual distribution misspecification however is countered by the Central Limit Theorem. As $N \longrightarrow \infty$ the estimated coefficients will be approximately normally distributed around the true coefficients, even if the distribution of log house prices is not normal (Greene, 2018) Hence, the vastness of the training set trivialises the normality requirement.

The random effects normality was similarly examined through a Q-Q plot (Figure 6). The random effects distribution is strongly non-normal, with the curve wildly deviating from the red identity line. Despite the failed random effects specification, the consequences on model fitting are likely to be unimportant. Verbeke and Lesaffre found that non-normal random effects still produce consistent and asymptotically unbiased results when assumed to be normally distributed (Verbeke G., 1997). The random effects coefficient estimates in this case were near zero for all sectors. Representing house-specific effects as random variables was primarily done to increase statistical efficiency, producing more precise parameter estimates, and reduce the computational burden of running the model. Consequently, we deem the failed assumption a non-issue.

To summarise, the ARME model appropriately specifies the autoregressive structure of the residual series. Although it does not perfectly model the heteroscedasticity in each sector-wide estimation, it is a vast improvement relative to the UP model.

Therefore, the relative improvement in RMSE by the ARME model is justified. Neither the residuals nor random effects are normally distributed. As explained above, the model is still asymptotically valid considering these findings. Hence, the diagnostics tests support the ARME estimation procedure.

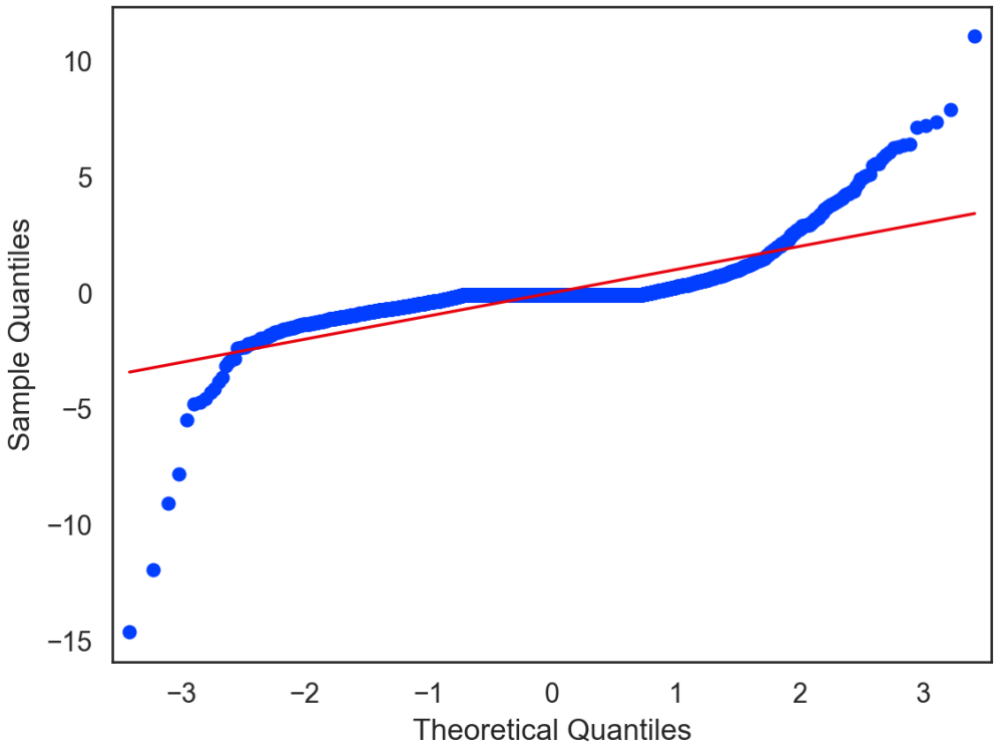

Figure 6: Random Effects Q-Q Plot

4.4 DISCUSSION

The ARME models differs from the UP model is two ways. Firstly, The ARME model specifies house specific effects as random effects and not fixed effects. The change is valid since house specific effects are likely independent of time effects. Secondly, the ARME model incorporates the serial correlation between sales of the same property and the heteroscedasticity arising from the time gap between repeat sales into its estimation. These changes primarily improve model efficiency and not estimator bias. The performance statistics verify this, with the ARME model scoring similarly in ME and MAE to the UP model, but vastly outperforming the model in RMSE.

The repeat-sales variant of the ARME model: ARME – RS, performed better than both the ARME and UP models in forecasting. The testing set however is unrepresentative of the overall UK housing market, as proven by the differing performance metrics obtained for the two mixed effects models. When comparing the three indices against the UK HPI for the entirety of England and Wales, the ARME model followed the hedonic index the closest.

The ARME estimates on average that if house price shock occurs at time t, 97.8% of it will persist into the next period. The average time between repeat sales is 6.7 years. Therefore, errors between adjacent sales of the same property share an average correlation of 0.861. The random effects estimates were essentially zero for all sector estimates, showing they have minimal predictive value in of themselves. The model diagnostics validate the use of the model for producing sound coefficient estimates. However, the model has limitations. The algorithm running the model only estimates the residual series of the model twice to find the values for ϕ and σ_ϵ^2 prior to running the main regression. Ideally, the process should be repeated until the parameter values converge within an acceptable threshold. This was not done for run-time reasons. As a result, close to 30% of sectors have autocorrelation values bordering on unity, a reality that is possible but highly unlikely given previous

findings (Hwang M., 2004), (Chaitra H. Nagaraja, 2011). Ultimately, we feel the ARME is superior to the UP model in predicative ability and market-wide representation.

Part II - Auction Discount Index

5 THEORETICAL BACKGROUND

The Auction Discount Index was first conceptualised in a paper from Fathom Consulting (Brigden, 2009). The idea is to map the discount or premium of auctioned property over time and compare this to movements in the 'conventional' housing market. The theory predicts that shocks to the value of auctioned property will foreshadow the trends in the conventional market. Here, the properties that make up the conventional market are bought and sold by private individuals using estate agents as an intermediary. The key difference between the selling of properties on the conventional market or by auction is the speed of sale. Auction sellers are generally either lenders holding repossessed properties or households facing financial distress (Corder M., 2010). These sellers are willing to sell the property below its value (discount) to get a quick sale. By contrast, conventional property sales may take months for a transaction to be registered due to buying chains or mortgage approvals for example. Sellers on the conventional market, not as desperate as auction sellers, want to maximise their sale price, resulting in a reluctance to readjust their reservation price to the current market conditions. Therefore, prices adjust faster to current market conditions on the auction market than the conventional market

In the second part of this project, we create the novel Auction Discount Index (ADI) by tracking the monthly median discount for properties sold by auction. The Essential Information Group (EIG) supplied us with property auction data for all properties sold on UK auction houses since 1991. Using the ARME valuation model developed in part I of this paper, we estimate what the properties in the EIG data would have sold for on the conventional market. The specifics on how we use this to create the ADI is in the next section. We then use Toda and Yamamoto's Augmented Granger causality Test (Toda H. Y., 1995) to verify that the ADI can help predict future trends in house price movements in section 7.2.

6 DATA AND METHODOLOGY

6.1 AUCTION SALES DATA

The final bid, address and date of auction sale were extracted for the approximately 307,000 successful auction sales listed in the EIG dataset. The details were fed into the ARME automated valuation model used in Section 4.2: Performance Testing to estimate the worth of the auctioned property according to the ARME model. Developing the ARME model in Part I was crucial for the construction of the Auction Discount Index. Most auction sellers probably tried and failed to sell the property on the conventional market. Therefore, the type of properties sold at auction are likely distinctly different from those listed on the HM Land Registry Price Paid dataset. And although the ARME model was trained using the Land Registry dataset, by incorporating single sales into its estimation it clearly suffered less from selection bias relative to the repeat-sale models.

The auction discount for a property is its final successful bid divided by its estimated worth at the time of auction. As explained in the training and performance section of part 1, the property valuation is found by inflating/deflating the closest previous (or future) house sale to estimate the value of the house at the specified date. Therefore, the property must feature on the Land Registry dataset at least once to find an estimate. Due to this, and the difficulty in parsing addresses to find the unique identifiers for the property, only around half of the dataset produced an estimated valuation. Nonetheless, 149,000 discount estimates over 324 months is still sizeable. Finally, the Auction Discount Index is defined by the median auction discount in each month across the sample period.

6.2 Toda and Yamamoto augmented Granger causality framework

Researchers commonly use Granger's (1969) causality test to investigate whether lagged values of a time series can help predict the future values of another series. The test works as follows:

Firstly, fit a Vector Autoregressive ,VAR(p) model for the two series ADI and HPI. This involves describing each variable as p lags of itself and p lags of the other time series, where p best specifies the underlying process of the variable in question:

$$HPI_t = \mu + \sum_{i=1}^{p} \alpha_i \, HPI_{t-i} + \sum_{i=1}^{p} \beta_i \, ADI_{t-i} + u_{1,t}$$

$$ADI_t = \mu + \sum_{i=1}^{p} \gamma_i \, ADI_{t-i} + \sum_{i=1}^{p} \delta_i \, HPI_{t-i} + u_{2,t}$$

(14)

Where μ is a constant, $\{\alpha, \beta, \gamma, \delta\}$ are coefficient vectors and the error terms u are assumed to be distributed $N \sim (0, \sigma^2)$. If lagged values of ADI aid in predicating future values of HDI the β coefficients would be jointly significant. Therefore, the following is tested:

$$H_0 : \beta_1 = \cdots = \beta_p = 0$$

(15)

If the Wald test χ^2 statistic exceeds the critical value at the 0.05 level of significance, the test concludes that ADI Granger-causes HDI.

The standard Granger-causality Framework however tends to give spurious results when the time series in question are integrated of different orders, mis specified or suffers from autocorrelation (Toda H. Y., 1994). The reason is because the VAR model follows the standard OLS assumptions, and under any of the above conditions the OLS parameters become inconsistent. Consequently, the obtained Wald test statistic is no longer asymptotically χ^2 distributed, invalidating the test.

In the next section, we show that ADI is stationary, but HDI is not. We could induce stationarity via differencing, but by doing so vital long run relationships between the variables will be lost. Additionally, we are primarily interested in whether ADI has any predictive power on HDI, not its first difference. Toda and Yamamoto (1995) propose an elegant and robust solution. The above VAR model is repeated using the levels of the series even if the series are non-stationary. The model however adds m lags to each variable in the system where m is equal to the maximum order of integration of the variables (eq.15). For example, if ADI is stationary but HDI is integrated of order 1, $m = 1$. The added m lags absorb the non-stationary dynamics from the model, ensuring the estimated coefficients for the first p lags are consistent, and the Wald test statistic is appropriately distributed. Therefore, the Granger causality test can proceed as usual.

$$HPI_t = \mu + \sum_{i=1}^{p+m} \alpha_i \, HPI_{t-i} + \sum_{i=1}^{p+m} \beta_i \, ADI_{t-i} + u_{1,t}$$

$$ADI_t = \mu + \sum_{i=1}^{p+m} \gamma_i \, ADI_{t-i} + \sum_{i=1}^{p+m} \delta_i \, HPI_{t-i} + u_{2,t}$$

(16)

The full Toda and Yamamoto procedure is as follows:

1. Perform the Augmented Dickey-Fuller (ADF) test both ADI and HPI to establish their respective orders of integration, with m equalling the maximum order.

2. Run equation (14) for $p = \{1,2 \dots 12\}$, extracting the Schwartz Criterion (SC), Akaike Criteria (AIC) and Hannan-Quinn Criterion (HQ) calculated for each VAR(p) model. The criteria measure how well the model fits the underlying data. The optimal lag length is chosen from the VAR(p) model minimising the information criteria.

3. Define the unrestricted VAR($p + m$) model.

4. Run the Lagrange Multiplier test for autocorrelation to ensure the model is void of autocorrelation.

5. Perform Wald test on the estimated coefficients for the first p lags of the model only to test for joint significance.

6. Conclude the result based on the obtained Wald test statistic

7 Results and Discussion

7.1 The auction discount index over time

The Auction Discount Index (ADI) is defined by the monthly median auction discount from 1996 to 2023 (Figure 7). A Five-month moving average was added for visual tractability. The auction discount generally hovers around 0.75 and 0.8 over the period shown. For comparison, the UK HPI is shown on the right of the figure. Looking from left to right, the index steadily rises to peak at 94% in 2004 and 92% in mid-2007. ADI increases are bullish market signals. Investors, expecting high future house market returns, flood property auctions to quickly build their portfolios. The higher demand pushes up auction bids, reducing the relative auction discount. The Auction Discount Index then plummets to 74% in 2009. Similarly, ADI drops are bearish signals. As more sellers fail to sell their properties conventionally, the more properties enter auction houses. The increased supply and seller desperation results in auction discounts to increase further. From 2009 to 2023 the ADI varies between 0.74 and 0.81.

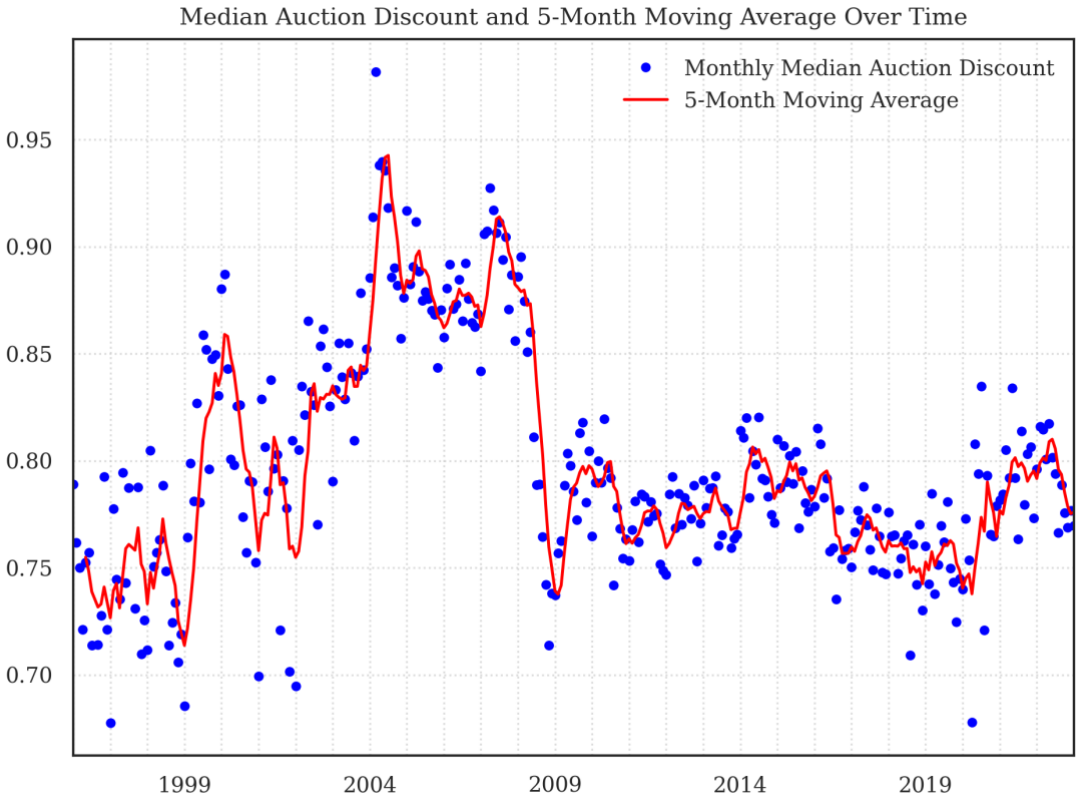

Figure 7: Auction Discount Index Over Time

7.2 GRANGER-CAUSALITY

Visual analysis is rarely enough to uncover the relationships between variables. Following the steps outlined previously, the ADF tests were run on the levels of ADI and HPI (Table 4). HPI failed to reject the null of stationarity, hence the test was repeated for its first difference. The first difference of HPI is stationary, hence $m = 1$.

Table 4: ADF tests

Variable	ADF-statistic	Prob.	Conclusion
ADI	-3.635	0.006***	Stationary
HPI	0.356	0.981	Non-stationary
ΔHPI	-4.590	0.000***	Stationary

Note: *** Indicates significance at the 0.01 level

Table 5 summarises the information criteria for each lag length up to $p = 12$. The lag length producing the lowest statistic is highlighted in bold for each criterion. AIC and HQ propose seven lags whereas SC proposes four. By majority rule, $p = 7$ was chosen. Therefore, the system of equations is defined by a VAR(8) process:

$$HPI_t = \mu + \sum_{i=1}^{8} \alpha_i \, HPI_{t-i} + \sum_{i=1}^{8} \beta_i \, ADI_{t-i} + u_{1,t}$$

$$ADI_t = \mu + \sum_{i=1}^{8} \gamma_i \, ADI_{t-i} + \sum_{i=1}^{8} \delta_i \, HPI_{t-i} + u_{2,t}$$

(17)

The LM test is conducted for 8 lags (Table 6). The null hypothesis is the absence of autocorrelation from the model. All 8 lags have p-values exceeding 10%. Therefore, we do not reject the null hypothesis and can continue to conduct the Granger-causality test.

Table 5: Information Criteria at different lag lengths

Lag, p	AIC	SC	HQ
0	0.174	0.199	0.184
1	-7.617	-7.544	-7.588
2	-7.819	-7.697	-7.770
3	-7.844	-7.673	-7.775
4	-8.148	**-7.929**	-8.060
5	-8.173	-7.905	-8.066
6	-8.189	-7.872	-8.062
7	**-8.234**	-7.868	**-8.088**
8	-8.214	-7.800	-8.048
9	-8.201	-7.738	-8.016
10	-8.225	-7.712	-8.020
11	-8.227	-7.666	-8.002
12	-8.232	-7.622	-7.988

Table 6: Lagrange Multiplier test for Autocorrelation results

Lag	Rao F-stat	Prob.
1	0.654014	0.6242
2	1.042529	0.3845
3	1.418387	0.2264
4	0.379846	0.8231
5	1.735485	0.1406
6	0.45336	0.77
7	0.654217	0.6241
8	0.917221	0.4534

Both Granger-causality of ADI on HPI and the reverse relationship are tested (Table 7). The null hypothesis in either test is that there is no Granger-causality. To be more specific:

Test 1: ADI → HPI

$$H_0: \beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = \beta_6 = \beta_7 = 0$$

(18)

Test 2: HPI → ADI

$$H_0: \delta_1 = \ \delta_2 = \delta_3 = \delta_4 = \delta_5 = \delta_6 = \delta_7 = 0$$

(19)

Test 1 overwhelmingly rejects the null hypothesis at the 0.01 significance level, concluding that past values of the ADI, up to 7 months previously, can help predict future values of HPI. The reverse however is not true looking the results of Test 2. The results confirm our hypothesis.

Table 7: Granger causality test

Test	Null Hypothesis:	χ^2	Prob.	Conclusion
1	ADI does not Granger Cause HPI	66.920	0.000***	Reject
2	HPI does not Granger Cause ADI	6.446	0.489	Not Reject

Note: *** indicates p-value < 0.01

8 CONCLUSIONS

In part I on this paper, we implemented Gao and Wang's (2007) Unbalanced Panel house price model and our own Autoregressive Mixed Effects model adapted from Nagaraja et al. (2011) paper. Our model attempted to reduce the selection bias inherent in repeat-sales models and rectify some of the oversimplistic assumptions of the UP model, such as house characteristics being time invariant and homoscedastic, autocorrelation free error terms. We model house specific effects as random variables, distributed $N\sim(0,\sigma_\tau^2)$ for each postcode sector. We also employ the Prais-Winsten transformation and GLS estimation procedure to appropriately model the structure of the error term. The model also uses an additional estimation equation to incorporate properties that have only sold once to partially remove the selection bias of the estimated indices.

Both models were estimated at the yearly postcode sector level for England and Wales before using cubic interpolation to obtain monthly indices. The ARME estimation results evidence slow mean reversion for exogenous house price shocks, with on average 97.8% of the shock remaining in the following year. The average gap between sales of the same property is 6.7 years, hence the average persistence between repeat-sales is 86.1%. Diagnostic testing confirmed that error heteroscedasticity and autocorrelation was sufficiently treated and the ARME model is valid. Prediction testing showed the ARME model was marginally beaten by the UP model in mean average error but significantly outperformed the model in root mean squared error. The tests however favoured repeat-sales methods. Consequently, the ARME was additionally implemented using only repeat-sales in its construction. The model outperformed the UP model in both MAE and RMSE. The three models were compared to the hedonic, highly specified UK HPI. The ARME followed the UK HPI the closest, concluding that homes that have sold multiple

times are distinctly different to homes that have not and the ARME model is more efficient and representative of the overall UK housing market than the UP model.

In part II we developed a new macroeconomic indicator, the Auction Discount Index, to help predict future house market movements. Properties sold via conventional means can take many months to finalise. The price of a property sold at auction is determined at the fall of the hammer. Hence, auction sales proxy the "mark-to-market" price, the value of the property if it was to sell today. As a result, the auction market adjusts faster to market conditions than the conventional property market. If the logic is sound, relative auction market price movements should lead conventional housing market movements. Consequently, we used the ARME model and auction sales data supplied by EIG to calculate the ratio between what the auctioned property sold for and its ARME estimate. The Auction Discount index was formed from the monthly median discount. Following, we employed Toda and Yamamoto's (1995) Augmented Granger-causality to confirm that for seven lags (months), the ADI predicts the current value of UK HPI better than when only using lags of UK HPI.

The positive results of the ARME model and ADI are of particular benefit to policymakers to ensure macroeconomic stability to the wider economy. We acknowledge the limitations of our ARME model. For convenience ϕ and σ_ϵ^2 were not necessarily iterated until convergence. Hence, the obtained coefficients may not be optimal. For future research, we recommend using the ARME to investigate regional housing trends. Additionally, after establishing Granger-causality between median auction discounts and house prices the next step is to measure the extent of ADI's predicative power.

9 References

Abraham, J. M. (1991). New evidence on home pricesfrom Freddie Mac repeat sales. *Real Estate Economics*, 333-352.

Agus, Q. M. (2015). Comparison Analysis of Time Series Data Algorithm Complexity for Forecasting of Dengue Fever Outcomes. *International Journal of Advanced Research in Computer Science.*

Allies Computing. (n.d.). *Address Data*. Retrieved from Allies computing: https://alliescomputing.com/address-data-101

Arthur Grimes, C. Y. (2010). *A Simple Repeat Sales House price index: Comparative Properties Under Alternative Data Genaration Processes.* Motu Economic and Public Policy Research.

Bailey, M. J. (1963). A regression method for real estate price index construction. *Journal of the American Statistical Associstion*, 58, 933-942.

Boham, H. N. (2016). The impact of regional commuter trains on property values: Price segments and income panel.

Breusch, T. S. (1979). A simple test for heteroskedasticity and random coefficient variation. *Econometrica*, 1287-1294.

Brigden, A. G. (2009, May 12). Valuing Houses.

Case, B. Q. (1991). The dynamics of real estate prices. *Rev. Econ. Statist.*, 50-58.

Case, K. E. (1987). Prices of single-family homes since 1970:New indexes for four cities. *N. Engl. Econ. Rev.*, 45-56.

Chaitra H. Nagaraja, L. D. (2011). AN AUTOREGRESSIVE APPROACH TO HOUSE PRICE MODELING. *The Annals of Applied Statistics*, 124-149.

Chi, E. M. (1989). Models for Longitudinal Data with Random Effects and AR(1) Errors. *Journal of the American Statistical Association*, 452-459.

Clapham, P. E. (2004). Revisiting the past: Revision in Repeat sales and Hedonic Indexes of House Prices. *Lusk Centre for Real Estate.*

Corder M., R. K. (2010). Residential property auction prices . Bank of England Quartely Bulletin.

Englund, P. Q. (1999). The choice of methodology for computing housing price indexes: Comparisons of temporal aggregation and sample definition. *Journal of real estate financial economics*, 91-112.

Gao, A. H. (2007). Multiple transactions model: A panel data approach to estimate housing market indices. *Journal of Real Estate Research* , 241-266.

Goetzmann, W. N. (1992). The accuracy of real estate indices:Repeat Sale estimators. *The Journal of Real Estate Finance and Economics*, 5-53.

Goetzmann, W. N. (1995). Non-temporal components of residential real estate appreciation. *Re. Econ. Statist*, 199-206.

Granger, C. W. (1969). Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrics*, 424-438.

Greene, W. H. (2018). *Econometric Analysis.*

Grimes, A. S. (2021). Repeat sales house price indices: comparative properties under alternative data processes. *New zealand Economic Papers* , 7-18.

Hausman, J. A. (1978). Specification Tests in Econometrics. *Econometrica*, 1251-1271.

HM Land Registry. (2022, November 22). *Quality and methodology*. Retrieved from GOV.UK: https://www.gov.uk/government/publications/about-the-uk-house-price-index/quality-and-methodology#methods-used-to-produce-the-UK-HPI

Hwang M., Q. J. (2004). Selectivity, Quality Adjustment and Mean Reversion in the Measurement of House values. *The Journal of Real Estate Finance and Economics*, 161-178.

J., G. C. (1969). Investigating Causal Relations by Econometric Models and Cross-Spectral Methods. *Econometrica*, 424-438.

Jiang, L. P. (2015). New methodology for constructing real estate price indices applied to the Singapore residential market. *Journal of Banking and Finance*, 1-11.

Meese, R. A. (1997). The construction of residential housing price indices: A comparison of repeat-sales, hedonic-regression, and hybrid approaches. *Journal of real estate financial economics*, 51-73.

Office for National Statistics. (2023, January). *UK House Price Index:January 2023*. Retrieved from ons.gov.uk: https://www.ons.gov.uk/economy/inflationandpriceindices/bulletins/housepriceindex/january2023#house-price-index-data

Palmquist, R. B. (1982). Measuring environmental effects on property values without hedonic regression. *Journal of Urban Economics*, 333-347.

Qilong, Y. (2020). Rndom Effects and AR(1) Model -- In Longitudinal Data Analysis. Graduate Department of Community Health, University of Toronto.

S&P Dow Jones Indices. (2023, September 26). *S&P CoreLogic case-Shiller Home price Indices*. Retrieved from S&P Global: https://www.spglobal.com/spdji/en/index-family/indicators/sp-corelogic-case-shiller/sp-corelogic-case-shiller-composite/#overview

Toda, H. Y. (1994). Vector autoregression and causality:a theoretical and simulation study. *Econometric review*, 259-285.

Toda, H. Y. (1995). Statistical Inference in Vector Autoregressions with possibly integrated processes. *Journal of Econometrics*, 225-250.

Verbeke G., L. E. (1997). The effect of misspecifying the random-effects distribution in linear mixed models for longitudinal data. *Computational Statistics and Data Analysis*, 541-556.

Wang, F. T. (1997). Estimating house price growth with repeat sales data: What's the aim of the game? *Journal of Housing Economics*, 93-118.

10 APPENDICES

10.1 APPENDIX A

UP and ARME models

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from scipy.sparse import csr_matrix
import warnings
warnings.filterwarnings(action='ignore', category=UserWarning, module='sklearn')
import statsmodels.api as sm
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt

traindata = pd.read_table(r'/Users/ali/Desktop/Fathom Project/code/lrdata_test.t
xt')

# Clean data so all sales for postcode sector/district can be easily found
lrdata2 = traindata.copy()
lrdata2['Price'] = lrdata2['Price'].astype('int32')
lrdata2['Year'] = lrdata2['Year'].astype('int16')
lrdata2['Month'] = lrdata2['Month'].astype('int8')
lrdata2 = lrdata2.dropna(subset = ['Postcode'])
lrdata2 = lrdata2.dropna(subset = ['PAON'])
lrdata2['PAON'] = lrdata2['PAON'].astype(str)
lrdata2['SAON'] = lrdata2['SAON'].astype(str)
lrdata2['Postcode'] = lrdata2['Postcode'].astype(str)
pd.to_datetime(lrdata2['Date'], yearfirst=True)
uniq_pc_district = lrdata2["PC District"].unique()
lrdata2.set_index(['PC District', "PAON", "SAON", "Street"], inplace=True)
lrdata2 = lrdata2.sort_index()
```

UP

All it is is: - $\ln(y_{i,t}) = M * \beta_t + A * \tau_i + \epsilon_{i,t}$

- M: Time fixed effect dummies
- A: Individual house fixed effect dummies
- y: Sale price

```
years = [str(year) for year in range(1995,2024)]
years = np.array(years).flatten()

# Define minimum transactions per postcode district. If under skip regression
min_transactions = 250
```

```
## to store avg gbp per post code per year
ss = pd.DataFrame(data= None, columns=uniq_pc_district, index=years)
## to store sales per post code per year
ss_n = pd.DataFrame(data= None, columns=uniq_pc_district, index=years)

rank_fail = []
ols_fail = []
progress = 0
""" Main Loop"""

# Count is here to see how many postcodes districts fail regression requirements
count_tran = 0
count_rank = 0
count_ols = 0

for dist in uniq_pc_district:

    progress +=1

    print(f"Running AVM for {dist}, ({progress}/{len(uniq_pc_district)})")

    area = lrdata2.loc[dist]
    area_reset = area.reset_index()

    # Drop single sales
    subset_cols = ["Type", "PAON", "SAON", "Street", "Town/City", "District", "C
ounty", "Postcode"]
    area_reset["Repeat"] = area_reset.duplicated(subset=subset_cols, keep=False)
    area1 = area_reset.copy()
    area1 = area1[area1.Repeat]

    # Skip district if sample size insufficient
    if len(area1) < min_transactions:
        count_tran += 1
        print(f" {dist} has Insufficient sample size")
        continue

    area1['idx'] = range(len(area1))

    # Vector of each unique property in area1
    uniq_prop = area1.drop_duplicates(subset = ["Type","PAON","SAON","Street","T
own/City","District","County", "Postcode"])
    uniq_prop_idx = np.array(uniq_prop['idx'], dtype = np.int32)
    uniq_prop_label = ["v"+str(prop) for prop in uniq_prop_idx]

    # Log price vector
    ln_p = np.array(np.log(area1['Price']))


# Matrix to store house specific effects.
    A = np.zeros(shape= (len(area1), len(uniq_prop['idx'])), dtype=np.int32)

    # Loop to place '1' in rows where sale related to that property, and 0 for a
ll other properties in the row
    counter = 0
    for i in range(len(area1)):
```

```
            if counter < len(uniq_prop_idx):
                if i == uniq_prop_idx[counter]:
                    counter +=1
            A[i][counter-1] = 1

        #Matrix for time effects
        M = pd.DataFrame(0, index = range(len(area1)), columns= years)

        for row in range(len(area1)):
            date = area1.iloc[row]['Year']
            M.iloc[row][str(date)] = 1

        del area, area1

        # X matrix combining A + M
        x = pd.DataFrame(A, columns= uniq_prop_label).join(M)
        # Remove any columns where all the values are 0
        x = x.loc[:, (x != 0).any(axis=0)]
        # Drop one Year-Month column to avoid linear dependency
        x = x.drop(columns=x.columns[len(uniq_prop_idx)])

        # Check to ensure X is full rank. If not, remove linearlly dependent columns
        if np.linalg.matrix_rank(x)!=  min(len(x), len(x.columns)):
            print("not full rank")
            U, s, Vt = np.linalg.svd(x, full_matrices=False)
            rank = (s > 0.0001).sum()
            independent_col_indices = np.abs(Vt[rank-1]) > 0.0001
            x = x.iloc[:, independent_col_indices]
            boo = independent_col_indices[:len(uniq_prop_idx)]
            uniq_prop_idx = np.delete(uniq_prop_idx, np.where(boo == False))

        # Convert to X to memory-efficient containter
        x_sparse = csr_matrix(x, dtype=float)
        x.columns = x.columns.astype(str)


        # Try and run the regression
        model = LinearRegression(fit_intercept=False)
        try:
            model.fit(x_sparse, ln_p)

            results = model.coef_
            print(f'OLS running for {dist} of length {len(x)}')


            # Vector of coefficients for properties and time effects
            B_i = results[:len(uniq_prop_idx)]
            B_t = results[len(uniq_prop_idx):]

            # Average house price estimate in absence of time effects for postcode d
istrict
            avgp_i = np.mean(np.exp(B_i))
            # Average house price for district over time
            avgp_t = np.exp(B_t)*avgp_i

            # Save results in dataframes
            ss[dist] = ss.index.map(dict(zip(x.columns[len(uniq_prop_idx):], avgp_t)
```

```
))
        ss_n[dist] = M.sum(axis=0)
    except Exception as e:
        if str(e) == "Factor is exactly singular":
            count_rank += 1
            rank_fail.append(dist)
            print(f'{dist} of length {len(x)} {str(e)}')
            ss[dist] = None
            ss_n[dist] = None
        else:
            count_ols += 1
            ols_fail.append(dist)
            print(f'OLS failed for {dist} of length {len(x)} due to {str(e)}')
            ss[dist] = None
            ss_n[dist] = None

# interpolate missing values
ss = ss.infer_objects(copy=False)
ss.interpolate(method='linear', axis=0, inplace=True)


print(f"{count_tran} distrcits failed minimum transactions requirement")
print(f"{count_ols} distrcits failed to run regression")
print(f"{count_rank} distrcits failed rank requirements")

#ss.to_csv(r'C:\Users\ali\Documents\AVM_folder\ss_dist12Sep.csv', index=True)
#ss_se.to_csv(r'C:\Users\ali\Documents\AVM_folder\ss_se_dist8Sep.csv', index=Tru
e)
#ss_n.to_csv(r'C:\Users\ali\Documents\AVM_folder\ss_n_dist12Sep.csv', index=True
)
```

ARME model

```
## dataframes to store estimated paramters
ss = pd.DataFrame(data= None, columns=uniq_pc_district, index=years)
ss_n = pd.DataFrame(data= None, columns=uniq_pc_district, index=years)
ss_phi_mu_msr = pd.DataFrame(data= None, columns=uniq_pc_district, index=['phi',
'mu', 'mrs'])
re_dict = {}

min_transactions = 250

ar1_fail = {}
mt_fail = []
full_rank_fail = []
not_converged = []

progress = 0


for dist in uniq_pc_district:
```

```
    progress +=1

    print(f"Running AVM for {dist}, ({progress}/{len(uniq_pc_district)})")

    ''' Step 0: Preprocessing step'''

    # Get all sales for current district
    area = lrdata2.loc[dist]
    area1 = area.copy()
    area1 = area1.reset_index()

    # Create unique property IDs to act as the i index
    area1['property_id'] = area1['PAON'].astype(str) + '_' + area1['SAON'].astyp
e(str) + '_' + area1['Postcode'].astype(str) + '!' + area1['Street'].astype(str)
+ '_' + area1['Type'].astype(str)
    area1 = area1.sort_values(by=['property_id', 'Date'])


    # Remove repeat sales of the same property within the same year as this mess
es up things
    area1 = area1.drop_duplicates(subset=['property_id', 'Year'], keep='first')


    # Number which sale of property i the row belongs to. Equivelent to j in the
paper
    area1['sale_number'] = area1.groupby('property_id').cumcount() + 1
    area1.reset_index(drop=True, inplace=True)

    # Skip this district/sector if the sample size is too small
    if len(area1) < min_transactions:
        mt_fail.append(dist)
        print(f"{dist} failed minimum transactions requirement")
        ss[dist] = None
        ss_n[dist] = None
        ss_phi_mu_msr[dist] = None
        continue

    # Create shifted versions of 'Year' and 'property_id' to see if the sale bel
ow belongs to same property as current in current row.
    # Used to calculate gamma
    area1['Next_Year'] = area1['Year'].shift(-1)
    area1['Next_property_id'] = area1['property_id'].shift(-1)

    # Calculate gamma only for same properties
    area1['gamma'] = np.where(area1['property_id'] == area1['Next_property_id'],
                              area1['Next_Year'] - area1['Year'], 0)

    # Create X and y
    X = pd.get_dummies(area1['Year'], dtype=float)
    X = X.drop(columns=X.columns[0])
    X.insert(0,'mu', 1.0)
    y = np.array(np.log(area1['Price']))

    N = len(y)
    # r is an N x N diagonal matrix that stores error term variances
    r = np.zeros((N,N))
```

```
    # Index of first sales, j = 1
    mask_first_sale = area1['sale_number'] == 1


    # diagonal terms where j = 1 are given 1s
    r[np.arange(N)[mask_first_sale], np.arange(N)[mask_first_sale]] = 1
    mask_same_property = area1['property_id'] == area1['Next_property_id']


    # Index for j > 1
    mask_subsequent_sales = ~mask_first_sale


    # The associated gamma is given in the previous sale of the property where j
>1
    gamma_pos = np.arange(N)[mask_subsequent_sales] - 1

    # Random effects are the individual properties
    groups = area1['property_id'].to_numpy()
    Z = pd.get_dummies(groups, dtype = float)
    unique_properties = Z.columns


    try:

        ''' Step 1. regress y on X to with group random effects to obtain estima
te of residuals '''

        model = sm.MixedLM(y, X, groups).fit(reml=False)

        resid0 = model.resid

        sigma_sqrd_eps = model.scale

        ''' Step 2. estimate phi from obtained residuals using non-linear least
squares '''
```

```
        def estimate_phi(residuals):

            # Extract residuals for sales j where j > 1  *** Only repeat sale
s have correlated errors ***
            resid_j = residuals[mask_subsequent_sales]

            # Extract the associated residuals for sales j-1
            resid_j_1 = residuals[gamma_pos]

            # Extract gamma for sales j where j > 1
            gamma_vec = area1['gamma'][gamma_pos].values

            #function to estimate phi from: u_j = (phi^gamma) * u_(j-1) + eps
            def nls_fun(resid_j_i, phi, gamma):
                fun_est =  resid_j_1 * np.power(phi,gamma)
                return fun_est
```

```
            # estimate phi using nls
            popt, pcov = curve_fit(lambda resid_j_1, phi: nls_fun(resid_j_1,
phi, gamma_vec), resid_j_1, resid_j, p0=[0.99], bounds = (0,1))

            phi_est = popt[0]
            return phi_est
```

```
        ''' Step 3: regress Ty on TX to obtian mu and beta estimates. Then repea
t regression in step 1 using these estimates to obtain residuals '''
```

```
def estimate_resid(phi, sigma_sqrd_eps):

            # create and populate T
            T = np.identity(N, dtype=np.float64)
            T[mask_subsequent_sales, gamma_pos ] = -np.power(phi, area1['gamm
a'][gamma_pos])

            # Diagonal element of r is 1-phi^(phi*gamma) if j > 1, and 1 othe
rwise.
            r[np.arange(N)[mask_subsequent_sales], np.arange(N)[mask_subseque
nt_sales]] = 1 - np.power(phi, 2 * area1['gamma'][gamma_pos])

            # Estimated variance-covariance matrix of the erros
            V = (sigma_sqrd_eps/(1-phi**2)) * r

            # P = estimated weights to premultiply to Ty and TX to correct fo
r the heteroscedasticity
            P = np.diag(1 / np.sqrt(np.diag(V)))

            Ty = T @ y
            TX = T @ X

            PTy = P @ Ty
            PTX = P @ TX

            # regress Ty on TX
            T_model = sm.MixedLM(PTy, PTX, groups)
            T_results = T_model.fit(maxiter = 1000)

            # sigma^2_eps estimate
            sigma2_eps = T_results.scale

            beta = T_results.params[:-1]
            re_effects = T_results.random_effects
            tau = np.array([re_effects[prop] for prop in unique_properties]).
ravel()
            Ztau = Z @ tau

            # calculate the residuals of the untransformed model using estima
ted beta, mu and tau
            residuals = y - X @ beta
```

```
            residuals = residuals - Ztau

            return residuals, sigma2_eps
```

```
        ''' Step 4: repeat steps 1 - 3 until phi converges '''
        while True:
            p_est = estimate_phi(resid0)
            resid0, sigma_sqrd_eps = estimate_resid(p_est, sigma_sqrd_eps)
            p_est2 = estimate_phi(resid0)
            break


        ''' Step 5: Estimate final paramter values with obtianed value of phi
and simga-squared.
            With phi and sigma-squared, the variance-covariance matrix of eps
can be defined. Therefore.
            the heteroscedasticity in eps can be corrected. The transformation
of y* = Ty and
            X* = TX allows for the removal autocorrelated errors, applying a
similar transformation
            to the Prais-Winston transformation.  '''

        phi = p_est2

        # Fill in variance-covaraince matrix diagonals
        r[np.arange(N)[mask_subsequent_sales],
np.arange(N)[mask_subsequent_sales]] = 1 - np.power(phi, 2 *
area1['gamma'][gamma_pos])

        # Transformation matrix to remove serial correlation
        T = np.identity(N, dtype=np.float64)
        T[mask_subsequent_sales, gamma_pos ] = -np.power(phi,
area1['gamma'][gamma_pos])

        # Var-cov matrix
        V = (sigma_sqrd_eps/(1-phi**2)) * r
```

```
        P = np.diag(1 / np.sqrt(np.diag(V)))

        Ty = T @ y
        TX = T @ X
        PTy = P @ Ty
        PTX = P @ TX

        # regress PTy on PTX
        fin_model = sm.MixedLM(PTy, PTX, groups)
        fin_results = fin_model.fit(reml=False, full_output = True,
maxiter=1000)

        sigma2_eps = fin_results.scale

        betas = fin_results.params[:-1]

        re_effects = fin_results.random_effects

        fitted_vals = fin_results.fittedvalues
        residuals = fin_results.resid

        msr = sigma2_eps + fin_results.cov_re
        mu = betas['mu']

        ss_phi_mu_msr[dist] = [phi,mu,msr]
        ss[dist] = betas[1:]

        # Extract random effects and save them in dictionary
        re = {}
        for key, value in fin_results.random_effects.items():
            new_key = "".join(key.split("!")[0])
            group_var = value.get('Group Var', None)
            re[new_key] = group_var
        re_dict[dist] = re

        converged = fin_results.converged
```

```
        print('****************************************************************')
        if converged == True:
            print(f"Model convergence successful for {dist} of length
{len(Ty)}")
        else:
            not_converged.append(dist)
            print(f"Model failed to converge for {dist} of length {len(Ty)}")
        print('****************************************************************')

    except Exception as e:
        ar1_fail[dist] = str(e)
        print(f'Estimation failed for {dist} of length {len(Ty)} due to
{str(e)}')
        ss[dist] = None
        ss_n[dist] = None
        ss_phi_mu_msr[dist] = None



ss = ss.infer_objects(copy=False)
ss.interpolate(method='linear', axis=0, inplace=True)



print(f"{len(mt_fail)} distrcits failed minimum transactions requirement")
print(f"{len(ar1_fail.keys())} distrcits failed to run regression")
print(f"{len(not_converged)} districts did not converge")
```

10.2 APPENDIX B

Splitting Training and Test sets

```
import numpy as np
import pandas as pd

# Load in full dataset
lrdata = pd.read_table(r'C:\Users\ali\Documents\AVM_folder\lrdata_test.txt')

# Clean dataset
lrdata2 = lrdata.copy()
lrdata2['Price'] = lrdata2['Price'].astype('int32')
lrdata2['Year'] = lrdata2['Year'].astype('int16')
lrdata2['Month'] = lrdata2['Month'].astype('int8')
lrdata2['SAON'] = lrdata2['SAON'].fillna('NA', inplace=True)
lrdata2['SAON'] = lrdata2['SAON'].astype(str)
lrdata2 = lrdata2.dropna(subset = ['Postcode'])
lrdata2 = lrdata2.dropna(subset = ['PAON'])
lrdata2['PAON'] = lrdata2['PAON'].astype(str)
lrdata2['Postcode'] = lrdata2['Postcode'].astype(str)

# find out how many properties in dataset
property_counts = lrdata2.groupby(['PAON', 'SAON', 'Postcode', 'Type']).size().r
eset_index(name='count')

# properties with 3+ sales
three_plus = property_counts[property_counts['count'] >= 3]

# final sale for 3+ properties
three_plus_fin = lrdata2[lrdata2.set_index(['PAON', 'SAON', 'Postcode']).index.i
sin(three_plus.set_index(['PAON', 'SAON', 'Postcode']).index)].groupby(['PAON',
'SAON', 'Postcode']).last().reset_index()

# properties with 2 sales
two_sales = property_counts[property_counts['count'] == 2]

# first sale of two
two_sales1 = lrdata2[lrdata2.set_index(['PAON', 'SAON', 'Postcode']).index.isin(
two_sales.set_index(['PAON', 'SAON', 'Postcode']).index)].groupby(['PAON', 'SAON
', 'Postcode']).last().reset_index()

# Find out how many fianl sales for properties sold only twice are needed to be
added to testset to reach 15%
how_much = int(0.15 * len(lrdata2) - len(three_plus_fin))
two_sales2 = two_sales1.sample(n=min(how_much, len(two_sales1)))

testset = pd.concat([three_plus_fin, two_sales2])
merged = pd.merge(lrdata2, testset, indicator=True, how='outer')
trainset = merged[merged['_merge'] == 'left_only'].drop('_merge', axis=1)
print(f"trainsett: {len(testset)}")
print(f"testset: {len(trainset)}")
print(f"Proportion of two sales properties in testset: {how_much/len(two_sales)}
")

#Save testset and trainset to path
```

10.3 Appendix C

Performance Testing

```
import numpy as np
import pandas as pd
import statistics
import math as m
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import MultipleLocator
import seaborn as sns
```

Import training set, testset and estimated house price indices for the models

```
trainset = pd.read_table(r'C:\Users\ali\Documents\AVM_folder\train_data.txt')
testset = pd.read_table(r'C:\Users\ali\Documents\AVM_folder\test_data.txt')

interp_index = pd.read_table(r'C:\Users\ali\Documents\AVM_folder\postcode_distri
cts_130923.csv', sep = ",")
ar_betas_m = pd.read_table(r'C:\Users\ali\Documents\AVM_folder\ss_ar_dist_betas0
410_interp.csv', sep = ",")
ar_params = pd.read_table(r'C:\Users\ali\Documents\AVM_folder\ss_ar_dist_params0
410.csv', sep = ",")
```

Clean the above csv files into pandas dataframes or dictionary objects to easily find information

```
uniq_dates = [[year+'-'+month for month in np.array(range(1,13)).astype(str)]for
year in np.array(range(1995,2024)).astype(str)]
uniq_dates = np.array(uniq_dates).flatten()
interp_index.columns = map(str.upper, interp_index.columns)
interp_index['DATEID01'] = uniq_dates
interp_index = interp_index.rename(columns={'DATEID01' : 'Index'})
interp_index = interp_index.set_index('Index')
interp_index_dict = {dist: interp_index[dist].to_dict() for dist in interp_index
.columns}

ar_betas_m.columns = map(str.upper, ar_betas_m.columns)
ar_betas_m['DATEID01'] = uniq_dates
ar_betas_m = ar_betas_m.rename(columns={'DATEID01' : 'Index'})
ar_betas_m = ar_betas_m.set_index('Index')
ar_params = ar_params.set_index('Unnamed: 0')
ar_betas_dict_m = {dist: ar_betas_m[dist].to_dict() for dist in ar_betas_m.colum
ns}
ar_params_dict = {dist: ar_params[dist].to_dict() for dist in ar_params.columns}

# converting columns to the minimum memeory time required to save memory
# setting index to unique property identifiers allows AVMs to find all property
# sales for a specific property quicky
lrdata = trainset.copy()
lrdata = lrdata.dropna(subset = ['Postcode'])
lrdata = lrdata.dropna(subset = ['PAON'])
lrdata['PAON'] = lrdata['PAON'].astype(str)
lrdata['Postcode'] = lrdata['Postcode'].astype(str)
lrdata['SAON'] = lrdata['SAON'].astype(str)
lrdata['Price'] = lrdata['Price'].astype('int32')
```

```
lrdata['Year'] = lrdata['Year'].astype('int16')
lrdata['Month'] = lrdata['Month'].astype('int8')
lrdata['dateYM'] = pd.to_datetime(lrdata['Year-Month'] + '-1', format='%Y-%m-%d'
)
lrdata.set_index(['PAON', 'SAON', 'Postcode'], inplace=True)
lrdata = lrdata.sort_index()
lrdata['dateYM'] = pd.to_datetime(lrdata['Year-Month'] + '-' '1', format='%Y-%m-
%d')

testset = testset.drop(testset.iloc[:, 4:13],axis = 1)
testset = testset.dropna(subset = ['Postcode'])
testset = testset.dropna(subset = ['PAON'])
testset['PAON'] = testset['PAON'].astype(str)
testset['Postcode'] = testset['Postcode'].astype(str)
testset['SAON'] = testset['SAON'].astype(str)
testset['Price'] = testset['Price'].astype('int32')
testset['Year'] = testset['Year'].astype('int16')
testset['Month'] = testset['Month'].astype('int8')
testset.head()
```

AVM values any address included in the land registry data at any specifed date using the estimater parameters using index inflation/deflation

```
def AVM_UP(number, postcode, when, name = None):
    when_y = int(when[:4])
    when_m = int(when[5:])
    t =  pd.to_datetime(when +'-' + '1', format='%Y-%m-%d')

    if name == None:
        name = 'nan'

    # Find all sales pertaining to the address using property number, postcod
e and name if provided

    try:
        sales = lrdata.loc[(number, name, postcode)]
        if sales.empty:
            return "NO LRDATA"
    except KeyError:
        return "NO LRDATA"

    sales = sales.copy()
    sales['distance'] = ((t.year - sales['dateYM'].dt.year) * 12  + t.month -
sales['dateYM'].dt.month)
    sales['distance'] = np.abs(sales['distance'])



    if sales.empty:
        return "NO LRDATA"

    # Extract closest sale for the property for the specified date
    sales = sales.loc[sales['distance'] == sales['distance'].min()]

    # Obtain time indices relevent to the sale
    dist, y_m, price = sales['PC District'].iat[0], sales['Year-Month'].iat[0
], int(sales['Price'].iat[0])
```

```
    p_i_when = interp_index_dict[dist][when]
    p_i_sale = interp_index_dict[dist][y_m]

    # Inflate/deflate previous sale price to date specified
    return price * (p_i_when / p_i_sale)

AVM_UP('22','TS7 0LN', '2002-3')
```

```
def AVM_ar_ym(number, postcode, when, name = None):
    when_y = int(when[:4])
    when_m = int(when[5:])
    t =  pd.to_datetime(when + "-1", format='%Y-%m-%d')

    if name == None:
        name = 'nan'

    try:
        sales = lrdata.loc[(number, name, postcode)]
        if sales.empty:
            return "NO LRDATA"
    except KeyError:
        return "NO LRDATA"



    sales = sales.copy()
    sales['distance'] = ((t.year - sales['dateYM'].dt.year) * 12  + t.month -
sales['dateYM'].dt.month)
    sales['distance'] = np.abs(sales['distance'])

    # This code is only included for the ADI construction since land registry
started to include auction sales
    # into the dataset
    sales = sales[sales['distance'] >= 6]

    if sales.empty:
        return "NO LRDATA"

    sales = sales.loc[sales['distance'] == sales['distance'].min()]

    dist, t_1, price, gamma = sales['PC District'].iat[0], sales['Year-Month'
].iat[0], np.log(int(sales['Price'].iat[0])), sales['distance'].iat[0] / 12

    if t_1 == when:
        return np.exp(price)

    # Inflate price using ARME model specification

    phi, mu, mrs = ar_params_dict[dist]['phi'], ar_params_dict[dist]['mu'], a
r_params_dict[dist]['mrs']

    beta_t = ar_betas_dict_m[dist][when]
    beta_t_1 = ar_betas_dict_m[dist][t_1]

    y_j = mu + beta_t + (phi**gamma) * (price - mu - beta_t_1)
```

```
    estimate = np.exp(y_j)

    return estimate

AVM_ar_ym('22','TS7 0LN', '2023-1')
```

Calculate performance metrics

```
ar_test1 = ar_test1.dropna(subset = ['Estimate'])
ar_test1 = ar_test1.loc[ar_test1['Estimate']!='NO LRDATA']
ar_test1 = ar_test1.loc[ar_test1['Estimate']!=0]
ar_test1['E'] = ar_test1['Estimate'].astype(float)-ar_test1['Price'].astype(floa
t)
ar_test1['AE'] = abs(ar_test1['E'])
ar_test1['SE'] = (ar_test1['E'])**2
ar_test1['PE'] = ar_test1['E']/ar_test1['Price']
ar_test1['PAE'] = abs(ar_test1['PE'])
ar_test1['PSE'] = (ar_test1['PE'])**2
ar_test2 = ar_test1.copy()
ar_test2.sort_values(by=['PAE'],ascending=False, inplace = True, ignore_index =
True)
ar_test2 = ar_test2.tail(-len(ar_test2)//20)

ar_yr_ds = {'ME': statistics.mean(ar_test2['E']), 'MAE': statistics.mean(ar_test
2['AE']), 'RMSE': m.sqrt(statistics.mean(ar_test2['SE'])), 'MPE': statistics.mea
n(ar_test2['PE']), 'MAPE': statistics.mean(ar_test2['PAE']), 'RMSPE': m.sqrt(sta
tistics.mean(ar_test2['PSE']))}
ar_yr_ds
```

Code for constructing the ADI

```
eig =  pd.read_table(r'C:\Users\ali\Documents\AVM_folder\EIG_data.csv', sep = ",
")

# Only keep relevent rows
eigdf = eig.drop(eig.columns[[0,1,4,5,10,11,12,14,15,16,17,18,19,41,42]], axis=1
)
eigdf.columns[18:24]
eigdf = eigdf.loc[(eigdf.iloc[:, 13] == 1) & (eigdf.iloc[:, 18:24] == 0).all(axi
s=1)]
eigdf = eigdf.loc[eigdf['LastBid']>0]
eigdf['Number'] = eigdf['FullAddress'].str.split(' ').str[0]

# Extract the house number and postcode from eig dataset for auction sale. Then
feed details into AVM for each row
# To eastimate the price of the property on the conventional market
df2 = eigdf
df2 = df2.copy()
df2['Estimate'] = df2.apply(lambda row: AVM_ar_ym(row.iloc[28], row.iloc[4], row
.iloc[8]), axis=1 )
df2 = df2.loc[df2['Estimate']!='NO LRDATA']
df2 = df2.dropna(subset = ['Estimate'])
df2['D/P'] = df2['LastBid'].astype(float)/df2['Estimate'].astype(float)
```

```
df2.to_csv(r'C:\Users\ali\Documents\AVM_folder\auction_estimated_ar.csv', index=
True)

df2['AuctionDate'] = pd.to_datetime(df2['AuctionDate'])
df2['AuctionDate'] = df2['AuctionDate'].dt.to_period('M')
# This is the ADI
median_group = df2.groupby('AuctionDate')['D/P'].median()

# Plot for Figure 7
```

```
moving_average = median_group.rolling(window=5).mean()
sns.set_theme(context='paper', style='white', palette='deep', font='DejaVu Se
rif', font_scale=1)
plt.figure(dpi=150)

median_group.plot(style='o', color='blue', label='Monthly Median Auction Disc
ount', markersize=3)
moving_average.plot(label='5-Month Moving Average', color='red')

plt.title('Median Auction Discount and 5-Month Moving Average Over Time')
plt.xlabel('')
plt.legend(frameon=False)
plt.grid(True, alpha=0.7, ls=':')
#sns.despine()
plt.box(True)
plt.tight_layout()

# x ticks in each year
years = np.arange(0, len(median_group), 12)
plt.gca().xaxis.set_major_locator(MultipleLocator(base=12))
plt.savefig('adi.png', dpi=300)
plt.show()
```

10.4 APPENDIX D

Table D.1

Heteroscedasticity tests for thirty different postcode sectors

	ARME		UP	
	χ^2	p-value	χ^2	p-value
MK15 9	0.919	0.587	5.324	**0.000**
SR6 0	2.324	**0.000**	7.621	**0.000**
CO6 1	1.030	0.421	2.979	**0.000**
B90 4	1.108	0.316	1.569	**0.000**
DY5 1	2.489	**0.000**	7.013	**0.000**
S65 1	1.891	**0.003**	42.291	**0.000**
PE7 3	1.444	0.061	15.471	**0.000**
SK10 2	0.907	0.606	16.641	**0.000**
SA6 5	0.788	0.778	4.995	**0.000**
NR2 2	1.210	0.206	10.437	**0.000**
WS6 7	1.136	0.283	56.011	**0.000**
GL52 3	1.413	0.074	3.456	**0.000**
BR3 4	1.275	0.151	-	-
L9 6	5.033	**0.000**	6.565	**0.000**
SW16 3	0.785	0.781	3.488	**0.000**
UB5 5	0.755	0.819	4.158	**0.000**
FY4 1	1.169	0.247	1.302	**0.000**
NE45 5	0.784	0.782	1.660	**0.000**
WR5 3	1.817	**0.005**	1.638	**0.000**

Note: bold values indicate heteroscedasticity test failed for postcode sector at the 0.05 significance level.
BR3 4 failed to run the UP regression due to the regressor matrix being singular.