M3 Challenge 2024:

A Tale of Two Crises: The Housing Shortage and Homelessness

TEAM #17868

March 4th, 2024

1 Executive Summary

To the Minister Rowley,

With rapid population growth in urban centers, homelessness has become an increasingly important issue for urban planners and city designers. For example, in Manchester, homelessness has more than tripled over 14 years [6]. As housing supply continues to rapidly decrease in relation to demand and prices continue to skyrocket, more and more people will begin to have to face homelessness as their only option, putting pressure on the social welfare programs that support these impoverished people. Therefore, it is of utmost importance to develop and pass remedial legislation to allow for more affordable housing for the future.

We predicted the number of total housing units in both Manchester and Brighton & Hove over the next 10, 20, and 50 years, corresponding to the years 2031, 2041, and 2071 respectively. Using a linear regression model, we used data from 1993-2022 to extrapolate future trends about the number of total housing units in these two cities. We found that housing is only set to increase to 137,400 units in 2031, 143,400 units in 2041, and 161,500 units in 2071 in Brighton & Hove. These numbers are 254,800, 274,100, and 332,100 units respectively for Manchester. This gradual increase in housing is a step in the right direction, but needs more help fro legislators to increase the rate of building.

Next, we created two sinusoidal-trend models to predict homelessness in Brighton & Hove and Manchester. Utilizing the time series data from 2008-2022, we predicted that over the next 10, 20, and 50 years the percentage of the population that is homeless would be 0.63%, 0.56%, and 0.47% for Brighton & Hove and 0.63%, 1.81%, and 3.03% for Manchester. These models gave us the ability to model the long-term behavior of homelessness in each of the desired regions.

We then conducted a thorough analysis of the myriad of factors that may contribute to homelessness in Brighton & Hove, building a heatmap of correlation coefficients to determine which factors most closely positively or negatively correlated with homelessness. Using these key determinants of homelessness, we utilized a multiple linear regression model in order to fit the homelessness data for Brighton & Hove for 2008-2022, giving us the ability to predict these same effects on homelessness given predictions on data such as population, housing disparity, etc.

We believe these results will assist you in determining the correct path forward to begin solving this issue for England's regions.

Contents

1 Executive Summary **1**

2 Global Assumptions **4**

3 Q1: It Was the Best of Times **4**

3.1 Defining the Problem 4
3.2 Assumptions 5
3.3 Variables 5
3.4 The Model 5
3.4.1 Developing the Model 5
3.4.2 Executing the Model 7
3.5 Results 7
3.6 Discussion 10
3.7 Sensitivity Analysis 10
3.8 Strengths and Weaknesses 11

4 Q2: It Was the Worst of Times **11**

4.1 Defining the Problem 11
4.2 Assumptions 11
4.3 Variables 12
4.4 The Model 12
4.4.1 Developing the Model 12
4.4.2 Executing the Model 13
4.5 Results 14
4.6 Discussion 15
4.7 Strengths and Weaknesses 15

5 Q3: Rising from the Abyss **16**

5.1 Defining the Problem 16
5.2 Assumptions 16
5.3 Variables 16
5.4 The Model 16
5.4.1 Developing the Model 16
5.4.2 Executing the Model 17
5.5 Results 18
5.6 Discussion 18
5.7 Strengths and Weaknesses 19

6 Conclusion **19**

References **21**

A Appendix **23**

A.1 Python Code for Question 1 23

A.2 Python Code for Question 2 31

A.3 Python Code for Question 3 38

2 Global Assumptions

1. **All housing, either occupied or vacant, will be considered as part of the housing supply.**
   - **Justification:** Data is provided for both vacant and occupied housing by the Mathworks Math Modeling Challenge, and both will be treated the same way, even in our analysis of homelessness. [6]
2. **Those who are eligible for homeless aid but are not unhoused will be neglected.**
   - **Justification:** The data given by the Mathworks Math Modeling Challenge Data Statement provides the count of those who are homeless with priority need, those who are homeless, but without priority need, and those who are eligible, but are not homeless. [6] We will only consider the first two categories, as those are the people who are currently true homeless
3. **COVID-19 will not affect housing supply after 2024**
   - **Justification:** Coronavirus has slowed considerably in the UK since its peak in 2020-2021 [5], therefore our model will not consider the previous impacts of COVID-19.
4. **Assumption: Zoning laws will not change drastically over the next 50 years**
   - **Justification:** One of the most significant roadblocks to new housing in the UK are zoning laws and regulations [14]. Hence, if these change, they could drastically slow down or increase production of new housing. However, for simplicity's sake, we will assume that no significant zoning laws will change over the next 50 years.

3 Q1: It Was the Best of Times

3.1 Defining the Problem

The question asks us to develop a model predicting housing supply in 2031, 2041, and 2071 in Brighton & Hove and Manchester. To this end, our model will consider past housing supply from these two regions.

3.2 Assumptions

1. **Assumption: Housing supply data from 1993-2021 is sufficient to determine housing supply data over the next 50 years**

    - **Justification:** The data provided by the Mathworks Math Modeling Challenge is from 1993 to 2021 [6], so we will assume that this period of time provides sufficient data for our model

3.3 Variables

Variable	Description	Unit
S_{bh}	Total housing units in Brighton & Hove	Housing Units
S_m	Total housing units in Manchester	Housing Units
t	Time	Years

3.4 The Model

3.4.1 Developing the Model

We chose a linear regression model to predict changes in housing data over time. Linear regression models are often used to predict future trends using present/past data, and in finance and academics [7], so they can apply to the housing market. Depending on the depth and accuracy of the training data, linear regression models are generally accurate across both short and long timeframes. Our data exhibits linearity (as shown in Figure 1), so we decided a linear regression model would be optimal.

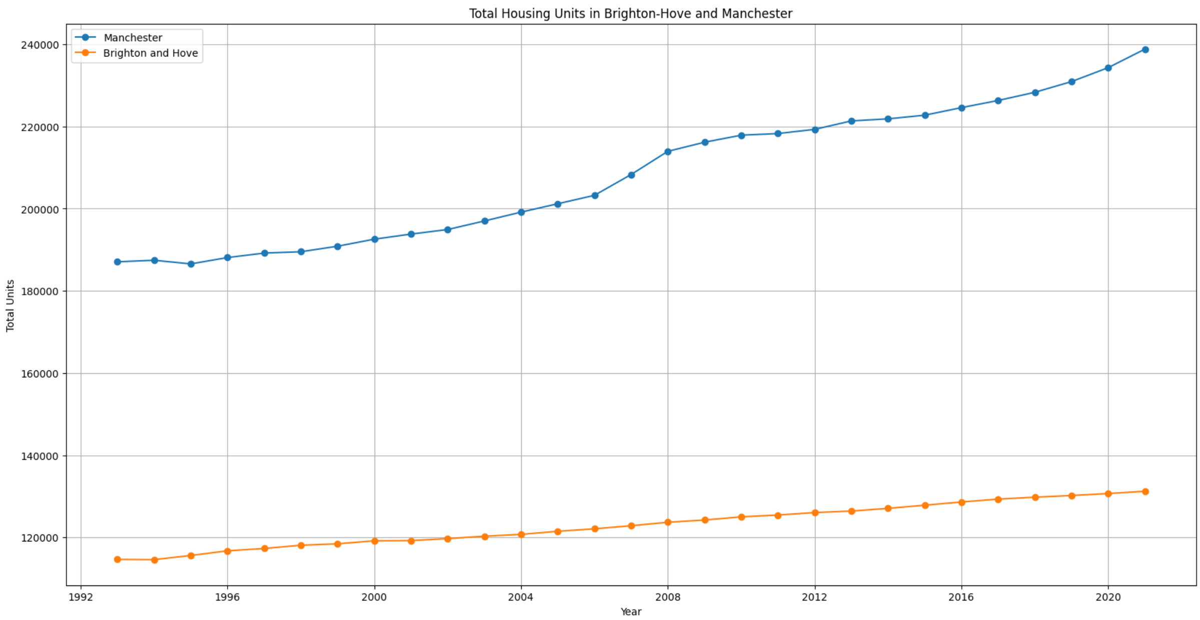


Figure 1: Housing data of Manchester (blue) and Brighton & Hove (orange) plotted over the years 1993-2021 [6].

A linear regression model develops a line of best fit that relates the variables x, y with an equation in the format $y = mx + b$. The coefficient m, which represents the slope of the line, and the constant b, which represents the y-intercept of the line. The values of m and b can be found with the equations below [8]:

$$m = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2} \tag{1}$$

$$b = \frac{(\Sigma y)(\Sigma x^2) - (\Sigma x)(\Sigma xy)}{n(\Sigma x^2) - (\Sigma x)^2} \tag{2}$$

Where:

- x is the years.
- y is the total housing units in the given city.

An Auto-Regressive Integrated Moving Average (ARIMA) model was also tested, as it would help accommodate for the slight non-linearity of the Manchester housing data and identify any potential seasonality and cyclical behavior in the data. The governing equation of ARIMA can be found below:

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t \tag{3}$$

Where:

- y'_t is the differenced time series value at time t. The prime notation indicates differencing, which is used to make the series stationary.

- c is a constant term that accounts for the mean level in the series.
- $\phi_1, \ldots, \phi_p$ are the coefficients of the autoregressive terms, where each term represents the impact of the p previous values of the time series on the current value.
- $y't-1, \ldots, y't-p$ are the p previous values of the differenced time series, indicating the autoregressive component of the model, which relates the current value of the series to its past values.
- $\theta_1, \ldots, \theta_q$ are the coefficients of the moving average terms, where each term represents the impact of the q previous forecast errors on the current value.
- $\epsilon_{t-1} \ldots, \epsilon_{t-q}$ are the q previous forecast errors. These errors are the differences between the past actual values and the past predicted values, indicating the moving average component of the model.
- ϵ_t is the error term at time t, representing the randomness or unpredictability in the series at time t. [3]

3.4.2 Executing the Model

To model the changes in housing supply, data that provided the total number of housing units available over a given time interval in each of the two regions was needed. Our team sourced the number of total housing units from the provided data statement [6]. The data provided in the data statement had the most complete and reliable data on the total number of housing units for a 30-year interval. While more data was preferable for more accurate predictions, a literature survey revealed no accessible sources that contained data as thoroughly cleaned and reviewed as that of the data statement.

After importing the data into our Jupyter notebook, we cleaned it and organized it into Pandas DataFrames. We used all of the provided data and used the *.polyfit()* function from NumPy to determine the linear regression model for each of the two regions' housing supplies. We set the *deg* parameter to one, as both city's housing unit data exhibited linearity. We then applied the linear regression model to predict the changes in housing supply by 2031, 2041, and 2071 (10, 20, and 50 years into the future from the most recent measurement year).

3.5 Results

The linear regression model we chose accounts for the lack of data and linear nature of the data provided. The deviations from the linear model in the Manchester housing data are very slight and are eventually balanced by deviations in the opposite direction, while there are no significant deviations from the linear model in the Brighton and Hove housing data. The results of the fit of the data provided by the model are seen in Figure 2.

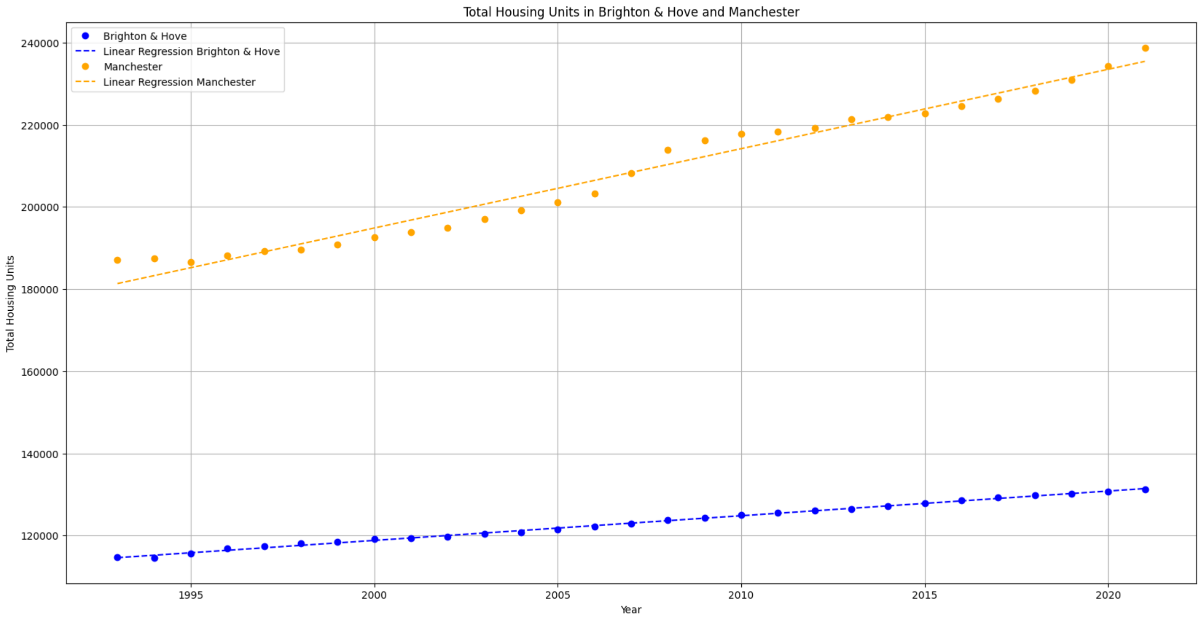


Figure 2: Line of best fit as predicted by linear regression model on housing data.

Next, the linear regression model was applied to future time frames (2031, 2041, and 2071 for 10, 20, and 50 years in the future, respectively). The predictions are shown in Figure 3.

The predictions were graphed with confidence intervals to ensuring that the model was reliable for making predictions. The narrow confidence interval suggests that a model has a tighter fit to the data, leading to more precise predictions. Moreover, the linear regression model had an $R^2 = 0.997$ for the Brighton & Hove housing data and an $R^2 = 0.974$ for the Manchester housing data, suggesting a tight fit to the given data. The future predictions with confidence intervals are seen in Figure 3

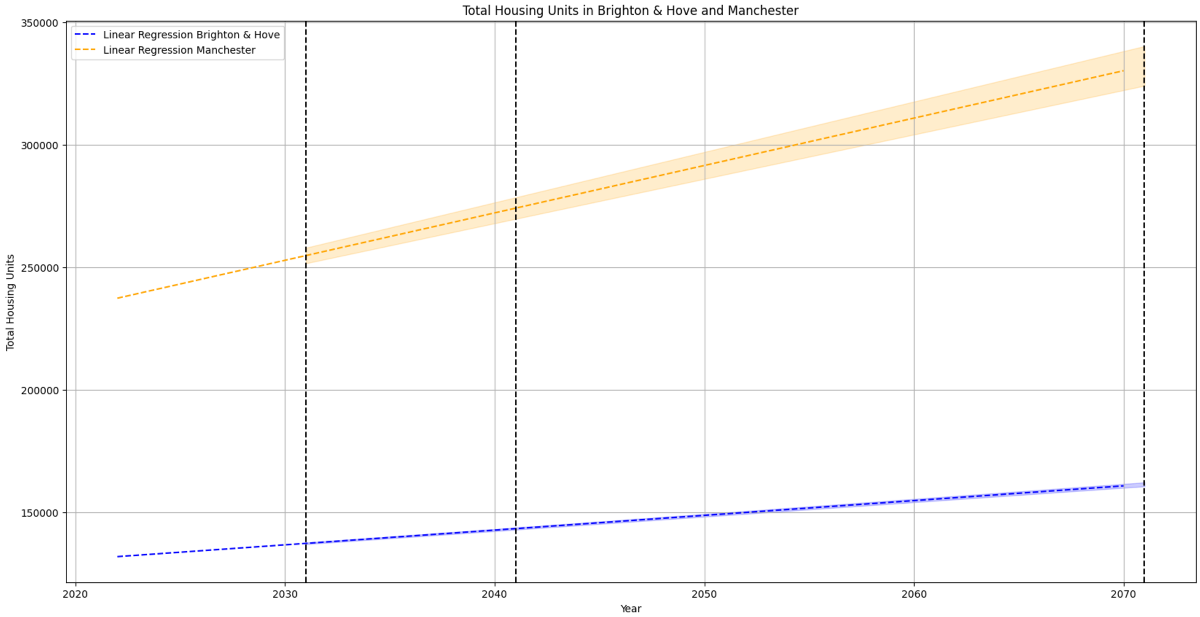


Figure 3: Predicted total housing units in Brighton & Hove and Manchester in 2031, 2041, and 2071. Confidence intervals are plotted in areas where predictions are made.

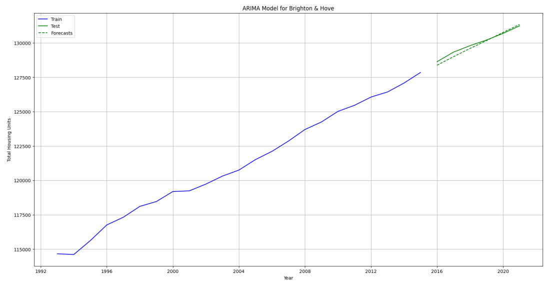
Year	Predicted Low	Predicted Average	Predicted High
2031	137,100	137,400	137,700
2041	143,000	143,400	143,900
2071	160,700	161,500	162,300

Table 1: Results for Brighton & Hove * Values are rounded to 4 sig-figs

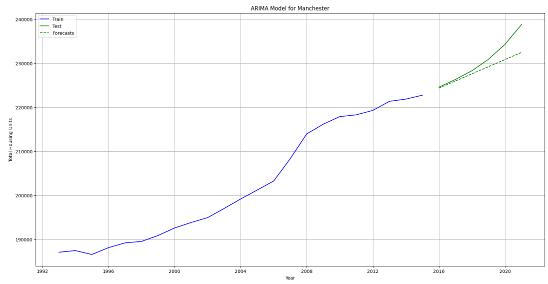
Year	Predicted Low	Predicted Average	Predicted High
2031	251,600	254,800	258,000
2041	269,700	274,100	278,500
2071	324,000	332,100	340,200

Table 2: Results for Manchester

The ARIMA model that was tested failed to properly model the data, as the volume of the data was insufficient to form a foundation upon which ARIMA could make predictions. Thus, the ARIMA model's predictions were linear in shape as well, but fit the data less accurately than the linear regression, as seen in Figure 4.



(a) Brighton & Hove housing data prediction using an ARIMA model



(b) Manchester housing data prediction using an ARIMA model

Figure 4: Testing an ARIMA Model

3.6 Discussion

In summary, our model predicts that there will be around 161,500 total housing units in Brighton & Hove and around 332,100 total housing units in Manchester by 2071. Hence, we can conclude that the number of total housing units in both regions will gradually grow, with Manchester's count increasing faster than than Brighton & Hove's.

3.7 Sensitivity Analysis

We performed a sensitivity analysis on our linear regression model by adjusting it's intercept and slope by the intervals seen below. The highest error was found only when adjusting the slope up to 1.05 while also adjusting the intercept down to 95% of its original value, or vice versa. When both the slope and intercept are adjusted in the same direction, the error is extremely low, only reaching a high of 5%.

Error in Slope	**Error in Intercept**								
	0.95			**1**			**1.05**		
	2031	**2041**	**2071**	**2031**	**2041**	**2071**	**2031**	**2041**	**2050**
0.95	-5.0	-5.0	-5.0	-44.4	-42.8	-38.5	-83.8	-80.5	-72.1
0.975	17.2	16.4	14.3	-22.2	-21.4	-19.3	-61.6	-59.1	-52.8
1	39.4	37.8	33.5	0.0	0.0	0.0	-39.4	-37.8	-33.5
1.025	61.6	59.1	52.8	22.2	21.4	19.3	-17.2	-16.4	-14.3
1.05	83.8	80.5	72.1	44.4	42.8	38.5	5.0	5.0	5.0

Table 3: Sensitivity Analysis of Slope and Intercept for Brighton & Hove Linear Regression

Error in Slope	Error in Intercept								
	0.95			1			1.05		
	2031	2041	2071	2031	2041	2071	2031	2041	2071
0.95	-5.0	-5.0	-5.0	-77.0	-71.9	-60.3	-149.1	-138.9	-115.5
0.975	33.5	31.0	25.1	-38.5	-36.0	-30.1	-110.5	-102.9	-85.4
1	72.0	66.9	55.3	0.0	0.0	0.0	-72.0	-66.9	-55.3
1.025	110.5	102.9	85.4	38.5	36.0	30.1	-33.5	-31.0	-25.1
1.05	149.1	138.9	115.5	77.0	71.9	60.3	5.0	5.0	5.0

Table 4: Sensitivity Analysis of Slope and Intercept for Manchester Linear Regression

Therefore, our model can be considered a reasonable predictor of total housing units in these two regions, but cannot guarantee predictions for new housing.

3.8 Strengths and Weaknesses

The linear regression model is the strongest model for the data given. The data for total housing units for both Brighton & Hove and Manchester both display a visually linear correlation over time, which allows for the linear regression model to excel at predicting future values. Other models that may have fit the model more appropriately, such as ARIMA, could not offer accurate predictions due to a lack of volume of data.

However, there are some weaknesses to the linear regression approach to the problem. It is possible that the subtler trends in the data, like seasonality and cyclical behavior, were overlooked by the linear regression model, thus ”smoothing over” potentially important details for future predictions. ARIMA and other generalized models potentially could account for these factors if given more in-depth data.

4 Q2: It Was the Worst of Times

4.1 Defining the Problem

The question requires developing a model that can predict the homeless population in 2032, 2042, and 2072 in Brighton & Hove and Manchester. The model must consider the past homeless population from these two region to forecast the future homeless population.

4.2 Assumptions

1. **The population growths of Manchester and Brighton & Hove are logistic**

   - **Justification:** The population of any general region can be modeled with an exponential function that plateaus at the region’s carrying capacity, the contributors to which are multivariate, thus following the logistic growth model [13]. It is assumed that Manchester and Brighton & Hove follow the pattern observed in many other places.

2. **Assumption: Homelessness data from the past is sufficient to determine future trends about homelessness**

   - **Justification:** The data provided by the Mathworks Math Modeling Challenge is only from 2008 to 2022, so we will assume that this data is not an outlier and is representative of the future trends.

3. **No major legislative change will take place**

   - **Justification:** The homeless issue in the UK, including Manchester and Brighton & Hove, is reaching crisis levels [9], so remedial legislation should and will most likely be passed. This will likely mean that homelessness never reaches the levels that any extrapolation from the present will predict, but since we cannot determine the effect of this potential future legislation, we must neglect this possibility.

4.3 Variables

Variable	Description	Unit
U_{bh}	Total homeless households in Brighton & Hove	Households
U_m	Total homeless household in Manchester	Households
P_{bh}	Total households in Brighton & Hove	Households
P_m	Total households in Manchester	Households
$U_{bh\%}$	Percent of households in Brighton & Hove that are homeless	%
$U_{m\%}$	Percent of households in Manchester that are homeless	%
t	Time	Years

4.4 The Model

4.4.1 Developing the Model

To predict homelessness in the regions of Brighton & Hove and Manchester for up to 50 years in the future, we developed two separate models that were tailored to the conditions of each city. For Brighton & Hove, we developed a sinusoidal regression fit, while for Manchester, we developed a sinusoidal fit with a trend.

Sinusoidal regression creates a curve of best fit that fits cyclical data with a sine function as shown in Equation 4. To extend this model further, a linear trend can be added as shown in Equation 5.

$$U_{bh\%} = \gamma sin(\zeta t + \phi) + \beta + \epsilon \tag{4}$$

$$U_{m\%} = \alpha * x + \gamma sin(\zeta t + \phi) + \beta + \epsilon \tag{5}$$

$$\tag{6}$$

where:

- α is the slope of the linear trend component.
- γ is the amplitude of the sinusoidal curve.
- ζ is the period of the sinusoidal curve.
- ϕ is the phase shift of the sinusoidal curve.
- β is the vertical shift of the sinusoidal curve.
- ϵ is the residual error. [2]

We also tested a correlation between an affordability score, which was calculated by dividing the median house sales price by inflation-adjusted median salary, and homelessness. However, this linear regression proved insignificant, as there was no correlation for Brighton & Hove, and the positive correlation determined for Manchester was notably weak, registering an R^2 coefficient of only 0.70.

Furthermore, we tested a correlation between the disparity between housing availability and overall population and homelessness in the two regions. However, for both regions, the correlation was weak at best, with an R^2 coefficient of 0.61 for Manchester.

We also tested a variety of models using the PyCaret AutoML library to fit multiple models to the data through machine learning.

4.4.2 Executing the Model

The data statement provided by the Mathworks Math Modeling Challenge [6] was used as the primary source of the data used for the model, as the other sources found did not have as extensive of a time range and as clean data for homelessness, total population, housing prices, and total housing unit count in the two regions as the data statement. This dataset did include some missing data; hence we linearly interpolated the value from the values around it in order to replace it.

To fit a sinusoidal curve to our data, we used the *.curve_fit()* function from the SciPy library. The function uses gradient descent as an optimizer and non-linear least squares as a loss function to find the curve of best fit in the form of any generalized function.

To add a trend to the sinusoidal fit, we combined the sinusoidal regression model used for the Brighton & Hove homelessness data and the linear regression model used for predicting future housing supply in Q1. By combining these, the sinusoidal fit followed an upward trend that better fit the data.

4.5 Results

Below is the table for our predictions of the percent of the population that will be homeless at 2032, 2042, and 2072. For reference, the last given data point from 2022 for the percentage of population facing homelessness was *0.53%* and *0.17%* for Manchester and Brighton & Hove, respectively.

Year	**Homeless Percent of Population**	
	Brighton & Hove (%)	**Manchester (%)**
2032	0.63	0.63
2042	0.56	1.81
2072	0.47	3.03

Table 5: Predicted Homeless Percentage of Population for Brighton & Hove and Manchester

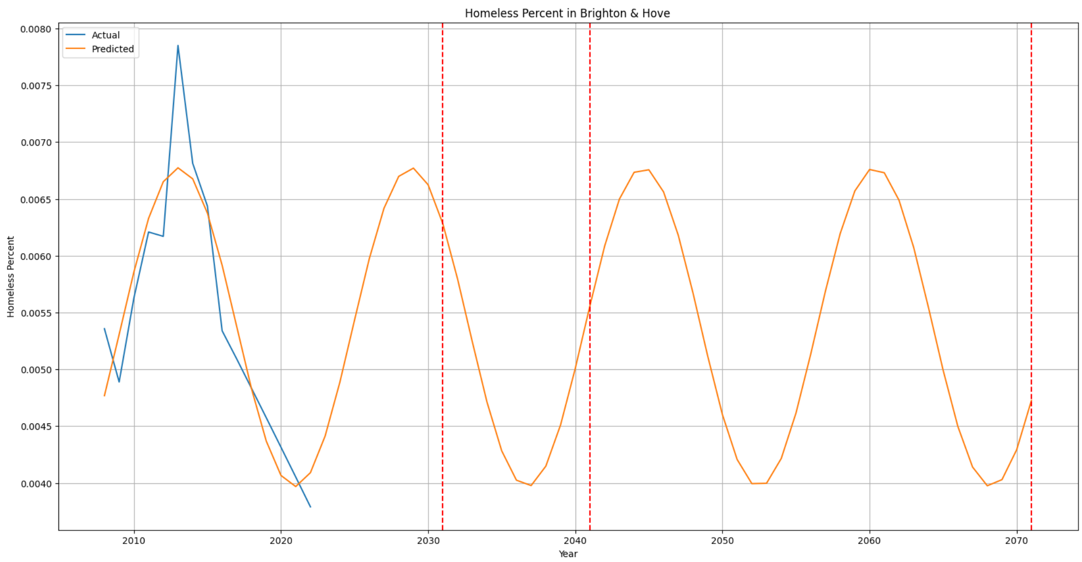


Figure 5: Extrapolating Sinusoidal Model to predict change in Homelessness in Brighton & Hove for the next 10, 20, and 50 years

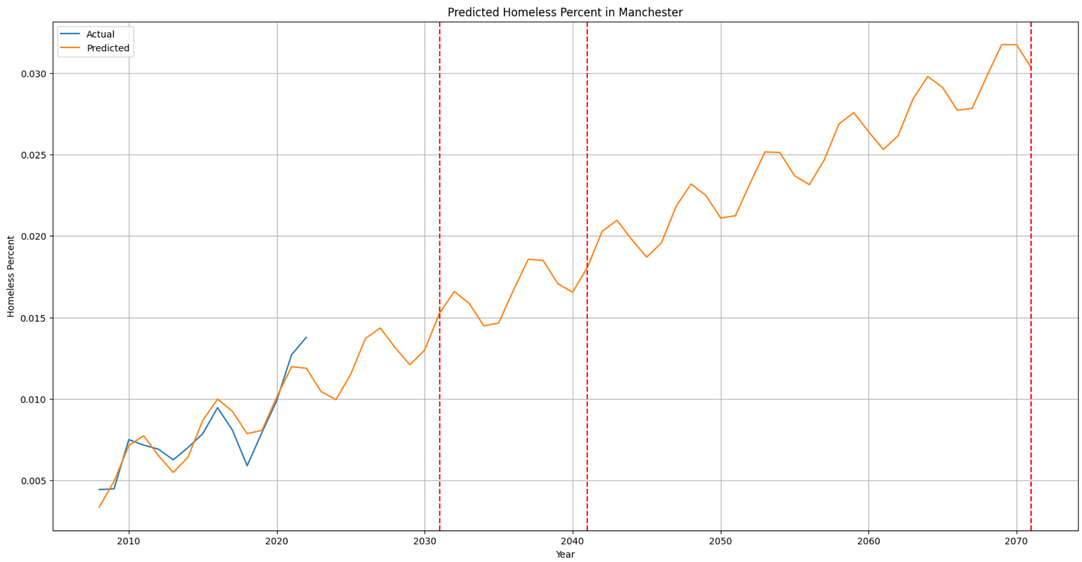


Figure 6: Extrapolating Sinusoidal Curve with Trend Model to predict change in Homelessness in Manchester for next 10, 20, and 50 years

4.6 Discussion

In summary, our model predicts that 0.63%, 0.56%, and 0.47% of the total population of Brighton & Hove will be homeless in 2032, 2042, and 2072, respectively. For Manchester, our model predicts that 0.63%, 1.81%, 3.03% of the total population will be homeless at the same time intervals. From this, we can conclude that if no remedial legislation is passed, homelessness will quickly become an unmanageable runaway issue in Manchester, and will linger as an economic and quality-of-life hamper in Brighton & Hove.

4.7 Strengths and Weaknesses

Both of our models' strengths lie in their specificity, as they focus only on the data of one city, rather than a general trend across both cities that would be less accurate to either. The models also focus specifically on the homelessness, rather than using correlations between other variables and homelessness, which could introduce confounding variables or interpret correlation as causation.

However, the lack of data could lead to issues in generalizability and overfitting. Considering the missing data in the Brighton & Hove set, it is possible that the trend fitted by the model would not truly extrapolate into the future.

5 Q3: Rising from the Abyss

5.1 Defining the Problem

The question asks us to develop a model that can predict the effects of unforeseen circumstances like economic recessions, or increased migrant populations to help a city determine a long-term plan and thus address homelessness. We assessed a myriad of different factors that contribute to homelessness to address the effects of these circumstances.

5.2 Assumptions

1. **Assumption: Predictions can be made from the past data available.**
   - **Justification:** The data provided by the Mathworks Math Modelling Challenge is only from 2018 to 2022, so we will assume for sake of simplicity that this time frame provides sufficient data for our model.
2. **Assumption: All variables will have an effect on the homeless population in Brighton & Hove.**
   - **Justification:** Most of the data provided in the data statement is related to income, housing availability, or population, which all can have an impact on the ability of an individual to secure a home.

5.3 Variables

Variable	Description	Unit
U_{bh}	Total homeless/unhoused households in Brighton & Hove	Households
P_{bh}	Total households in Brighton & Hove	Households
T_{bh}	Total population in Brighton & Hove	People
t	Time	Years
A_m	Median Age	Years
I_m	Median Income	Pounds
H_d	Housing Deficit/Disparity	Housing Units

5.4 The Model

5.4.1 Developing the Model

To identify the core factors impacting homelessness in a given city, we utilized a correlation heat map to determine the explained variance each factor contributed. This approach not only led to the identification of key determinants but also provided insights into the interplay between different factors. Some factors that we tested that didn't show any significant correlation to homelessness or other significant contributors included increase in population, England's total homeless population, and the average household size. As we were looking for factors that either significantly positively or negatively correlated with homelessness, signified by a value near *1* or *-1*, these factors did not exhibit a high enough correlation coefficient to be included at all within the multiple linear regression.

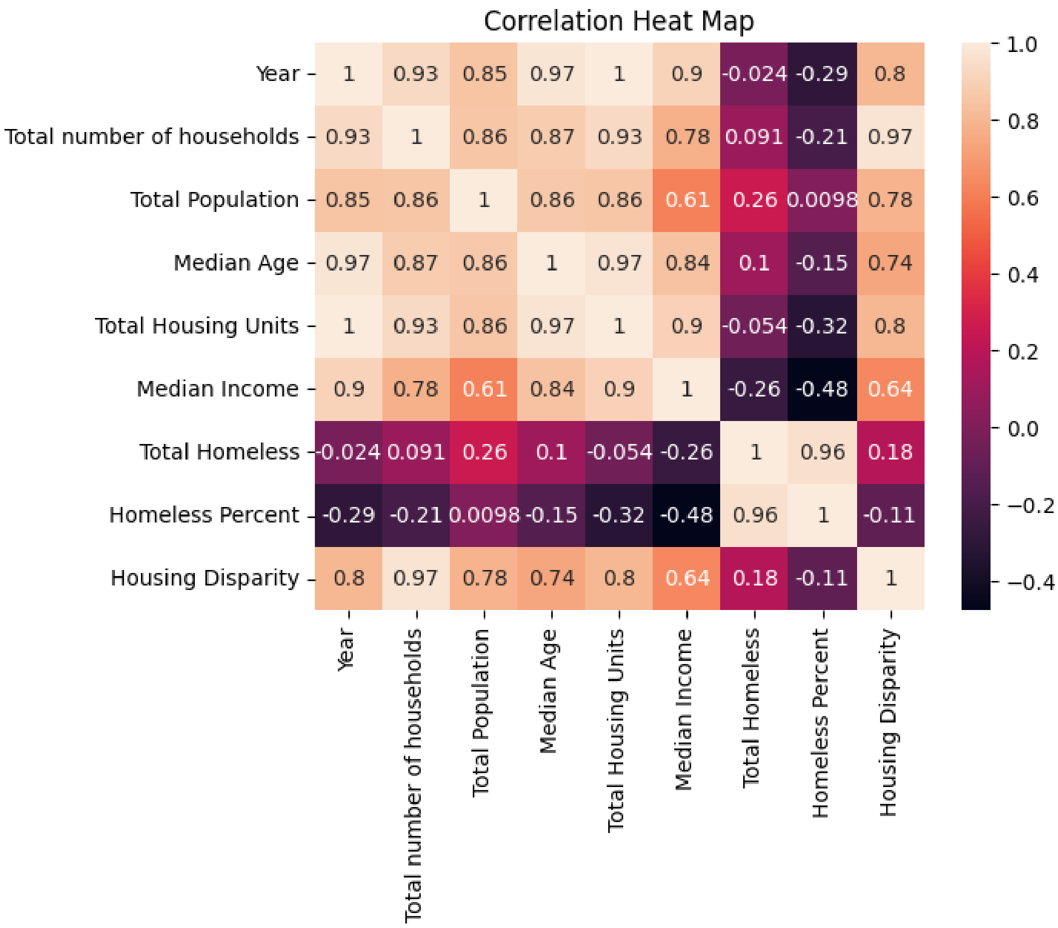


Figure 7: Heatmap of correlation between multiple factors contributing to homelessness in Brighton & Hove.

5.4.2 Executing the Model

To model the changes in homelessness, we gathered all our Brighton & Hove data for 2008-2021 from the provided data statement [6]. We utilized a multiple linear regression using the Python library SciPy. To address missing data, we decided to use linear interpolation to derive them from the surrounding points due to the minimal number of data samples.

5.5 Results

Figure 8 is the graph for our multiple regression model's fit to the homelessness data given for Brighton & Hove. As can be seen, the multiple linear regression fit is strong with an $R^2 = 0.85$.

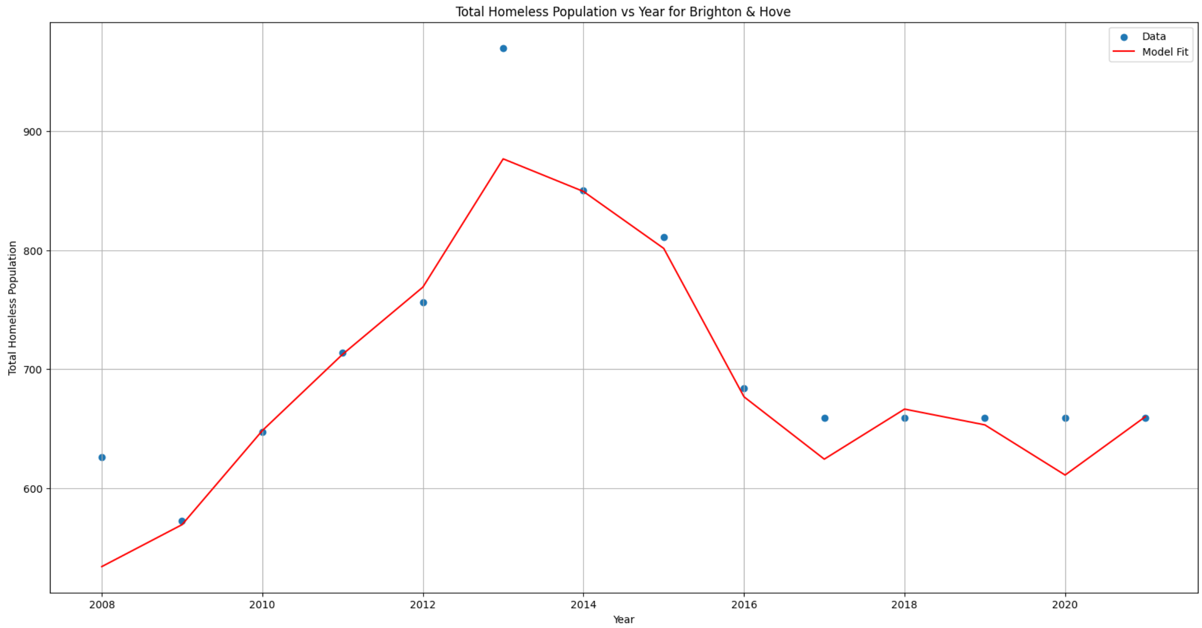


Figure 8: Line of best fit to Brighton & Hove's homeless data as predicted by multiple linear regression model on housing data.

To forecast future changes in homelessness, we plan to develop individual time-series models for each of the various factors. Since each of these individual models can be combined to produce an overall result for homelessness as shown in Equation 7, our resulting model is very flexible and dynamic.

$$U_b'h(t) = P_b'h(t) + T_b'h(t) + A_m'(t) + I_m'(t) + H_d'(t) + \epsilon \tag{7}$$

5.6 Discussion

The model predicts that housing disparity, as defined in Q2, and median income are the main factors that affect a region's homeless population. A significant disparity between a region's housing availability and its total population, coupled with a low median income, creates a breeding ground for a larger homeless population. When the number of available dwellings falls short of the city's total residents, competition for housing intensifies. This, in turn, drives up housing costs, making it increasingly difficult, especially for low-wage earners, to afford rent or mortgage payments. As a result, individuals and families who fall behind on payments or face unexpected financial hardships are more likely to be evicted or forced to live in doubled-up situations, further straining available resources. This lack of affordable housing options, combined with a low median income that makes saving for emergencies challenging, pushes vulnerable individuals and families towards homelessness.

The model also predicts that the total population of a region and the homeless population of the region are correlated. As the population of a city rises, its homeless population will also naturally rise as the housing disparity increases.

City legislators who want to develop a long-term plan to address homelessness must, over all else, increase the amount of available housing to combat homelessness. The housing disparity can be decreased in two ways: by reducing the population of the city and by increasing the number of available housing units. However, decreasing the population of a city can prove detrimental to the welfare of the city due to decreases in gross domestic product and thus the economic well-being of the city. By increasing the number of available housing units, however, legislators can also drive down the cost of housing, which addresses another key cause of homelessness, a lack of real estate purchasing power.

There are many outside shocks that could potentially disrupt an ideal situation and drive up homelessness. These can be accounted for in our model with variables such as median income and total housing units, whose downturn in a natural disaster or economic recession would affect homelessness in the region.

5.7 Strengths and Weaknesses

Since each of these individual models is combined together to produce an overall result for homelessness, our resulting model is very flexible and dynamic. With more data, we can easily tune and extend this model with additional factors that we find with more research.

However, the most significant weakness of this model is that it cannot fit well onto nonlinear input features as derived from its various sub-models. However, this weakness can be compensated by the fact that the sub-models themselves can be nonlinear.

6 Conclusion

For the first question, we used linear regression to find the total number of housing units available multiple decades into the future. Our model predicted 137,400 units in 2031, 143,400 units in 2041, and 161,500 units in 2071 in Brighton & Hove, and 254,800 units in 2031, 274,100 units in 2041, and 332,100 units in 2071 in Manchester. We used sinusoidal regression with trends to model the homelessness in the two regions over 50 years, finding 0.63% homelessness in 2032, 0.56% homelessness in 2042, and 0.47% homelessness in 2072 in Brighton & Hove, and 0.63% homelessness in 2032, 1.81% homelessness in 2042, and 3.04% homelessness in 2072 in Manchester. Finally, we used a multivariate linear regression model to examine the effects of 9 variables on homelessness in Brighton & Hove, allowing us to determine that the disparity in housing units available and total population in the region and the median income of the region had the greatest impact on the homeless population of the region.

References

[1] Nyc.gov, 2023. URL: https://comptroller.nyc.gov/reports/accounting-for-asylum-seekers/.

[2] *2.8 Phase Shifts and Sinusoidal Curve Fitting.* URL: https://www.cerritos.edu/dford/SitePages/Math_140/Math140Lecture8b_sp07.pdf.

[3] *9.8 Forecasting — Forecasting: Principles and Practice (3rd ed).* URL: https://otexts.com/fpp3/arima-forecasting.html (visited on 03/05/2024).

[4] Cassie Barton, Wendy Wilson, and Lorna Booth. "Tackling the under-supply of housing in England." In: *commonslibrary.parliament.uk* (May 2023). URL: https://commonslibrary.parliament.uk/research-briefings/cbp-7671/.

[5] *COVID-19 travel rules: Remaining restrictions for entry to UK, including passenger locator forms, will end this week.* Sky News. URL: https://news.sky.com/story/covid-19-travel-rules-remaining-restrictions-for-entry-to-uk-including-passenger-locator-forms-will-end-this-week-12566228.

[6] *https://m3challenge.siam.org/kdfrldh/.* M3 Challenge. URL: https://m3challenge.siam.org/kdfrldh/.

[7] IBM. *About Linear Regression — IBM.* www.ibm.com, 2023. URL: https://www.ibm.com/topics/linear-regression.

[8] *Linear Regression-Equation, Formula and Properties.* BYJUS. URL: https://byjus.com/maths/linear-regression/.

[9] George Lythgoe. *'Crisis and destitution' could be future for one Greater Manchester town.* Manchester Evening News, Mar. 2024. URL: https://www.manchestereveningnews.co.uk/news/crisis-destitution-homelessness-could-future-28727009 (visited on 03/05/2024).

[10] Kostas Mouratidis. "Urban planning and quality of life: A review of pathways linking the built environment to subjective well-being." In: *Cities* 115 (Aug. 2021), p. 103229. DOI: 10.1016/j.cities.2021.103229.

[11] Engineering National Academies of Sciences. *Permanent Supportive Housing: Evaluating the Evidence for Improving Health Outcomes Among People Experiencing Chronic Homelessness.* July 2018. URL: https://nap.nationalacademies.org/catalog/25133/permanent-supportive-housing-evaluating-the-evidence-for-improving-health-outcomes.

[12] Shelter. *At least 271,000 people are homeless in England today.* Shelter England, Jan. 2023. URL: https://england.shelter.org.uk/media/press_release/at_least_271000_people_are_homeless_in_england_today.

[13] Gilbert Strang and Edwin Herman. *8.4: The Logistic Equation*. Mathematics LibreTexts, July 2016. URL: https://math.libretexts.org/Bookshelves/Calculus/Calculus_(OpenStax)/08%3A_Introduction_to_Differential_Equations/8.04%3A_The_Logistic_Equation.

[14] *Why Britain doesn't build - Works in Progress*. worksinprogress.co. URL: https://worksinprogress.co/issue/why-britain-doesnt-build/.

A Appendix

A.1 Python Code for Question 1

```python
# %%
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
import statsmodels.api as sm

# Load the data
# Go one folder back
os.chdir("..")
path = "C:\\Users\\sapta\\OneDrive\\Documents\\GitHub\\M3Challenge2024\\data\\bh_housin
bh_housing = pd.read_csv(path)

# Convert the Year column to datetime
bh_housing['Year'] = pd.to_datetime(bh_housing['Year'], format='%Y')

# Convert all other columns to numeric.
for col in bh_housing.columns[1:]:
    if bh_housing[col].dtype == 'object':
        # Remove commas and convert to numeric
        bh_housing[col] = pd.to_numeric(bh_housing[col].str.replace(',', ''))

print(bh_housing.head())

# %%
path = "C:\\Users\\sapta\\OneDrive\\Documents\\GitHub\\M3Challenge2024\\data\\mancheste
manchester_housing = pd.read_csv(path)

# Convert the Year column to datetime
manchester_housing['Year'] = pd.to_datetime(manchester_housing['Year'], format='%Y')
```

```python

# Convert all other columns to numeric.
for col in manchester_housing.columns[1:]:
    if manchester_housing[col].dtype == 'object':
        # Remove commas and convert to numeric
        manchester_housing[col] = pd.to_numeric(manchester_housing[col].str.replace(',', ''))
print(manchester_housing.head())

# %%
plt.figure(figsize=(20, 10))
plt.plot(manchester_housing['Year'], manchester_housing['Total housing units'], '-o', label='Manchester')
plt.plot(manchester_housing['Year'], bh_housing['Total housing units'], '-o', label='Brighton & Hove')
plt.xlabel('Year')
plt.ylabel('Total Units')
plt.grid()
plt.legend(loc='upper left')
plt.title('Total Housing Units in Brighton & Hove and Manchester')
plt.show()

# %%
# Plot the data for the two cities
plt.figure(figsize=(20, 10))

# Make a linear regression using numpy.
m_bh, b_bh = np.polyfit(bh_housing['Year'].dt.year, bh_housing['Total housing units'], 1)
print('Brighton & Hove', 'm:', m_bh, 'b:', b_bh)

# Calculate the r^2 value
r_squared = np.corrcoef(bh_housing['Year'].dt.year, bh_housing['Total housing units'])[0, 1]**2
print(f"R^2: {r_squared}")

plt.plot(bh_housing['Year'].dt.year, bh_housing['Total housing units'], 'o', label='Brighton & Hove', color='blue')
```

```python
plt.plot(bh_housing['Year'].dt.year, m_bh*bh_housing['Year'].dt.year +
    b_bh, '--', label='Linear Regression Brighton & Hove', color='blue')

# Make a linear regression using numpy.
m_manchester, b_manchester =
    np.polyfit(manchester_housing['Year'].dt.year,
    manchester_housing['Total housing units'], 1)
print('Manchester', 'm:', m_manchester, 'b:', b_manchester)

# Calculate the r^2 value
r_squared = np.corrcoef(manchester_housing['Year'].dt.year,
    manchester_housing['Total housing units'])[0, 1]**2
print(f"R^2: {r_squared}")

plt.plot(manchester_housing['Year'].dt.year, manchester_housing['Total
    housing units'], 'o', label='Manchester', color='orange')
plt.plot(bh_housing['Year'].dt.year,
    m_manchester*bh_housing['Year'].dt.year + b_manchester, '--',
    label='Linear Regression Manchester', color='orange')

plt.xlabel('Year')
plt.ylabel('Total Housing Units')
plt.grid()
plt.legend(loc='upper left')
plt.title('Total Housing Units in Brighton & Hove and Manchester')
plt.show()

# %%
# Create statsmodels linear regression for Brighton & Hove
# Use statsmodels to calculate the confidence interval.
alpha = 0.05 # 95% confidence interval
bh_lr = sm.OLS(bh_housing['Total housing units'],
    sm.add_constant(bh_housing['Year'].dt.year)).fit()
print('Brighton & Hove')
print(bh_lr.summary())

# Create statsmodels linear regression for Manchester
# Use statsmodels to calculate the confidence interval.
alpha = 0.05 # 95% confidence interval
```

```
manchester_lr = sm.OLS(manchester_housing['Total housing units'], sm.add_constant(manchester_housing['Year'].dt.year)).fit()
print('\n\n\nManchester')
print(manchester_lr.summary())

# Extrapoloate the data for the next 10, 20, 50 years.
# Create confidence intervals for the next 10, 20, 50 years.
years = np.array([2031, 2041, 2071])
bh_conf_interval = bh_lr.get_prediction(sm.add_constant(years)).conf_int(alpha=alpha)
bh_housing_pred_low = bh_conf_interval[:, 0]
bh_housing_pred_average = bh_lr.predict(sm.add_constant(years))
bh_housing_pred_high = bh_conf_interval[:, 1]
print('Brighton & Hove')
for year, pred_low, pred_avg, pred_high in zip(years, bh_housing_pred_low, bh_housing_pred_average, bh_housing_pred_high):
    print(year, pred_low, pred_high)

# Make a table for Brighton & Hove
pred_bh = pd.DataFrame({'Year': years, 'Low': bh_housing_pred_low, 'Average': bh_housing_pred_average, 'High': bh_housing_pred_high})
print(pred_bh)

manchester_conf_interval = manchester_lr.get_prediction(sm.add_constant(years)).conf_int(alpha=alpha)
manchester_housing_pred_low = manchester_conf_interval[:, 0]
manchester_housing_pred_average = manchester_lr.predict(sm.add_constant(years))
manchester_housing_pred_high = manchester_conf_interval[:, 1]
print('\nManchester')
for year, pred_low, pred_avg, pred_high in zip(years, manchester_housing_pred_low, manchester_housing_pred_average, manchester_housing_pred_high):
    print(year, pred_low, pred_avg, pred_high)

# Make a table for Manchester
```

```
pred_manchester = pd.DataFrame({'Year': years, 'Low': manchester_housing_pred_low, 'Average': manchester_housing_pred_average, 'High': manchester_housing_pred_high})
print(pred_manchester)

# Plot the predicted data for the two cities.
time = np.arange(2022, 2071, 1)
plt.figure(figsize=(20, 10))
plt.plot(time, m_bh*time + b_bh, '--', label='Linear Regression Brighton & Hove', color='blue')
plt.fill_between(years, bh_housing_pred_low, bh_housing_pred_high, color='blue', alpha=0.2)
plt.plot(time, m_manchester*time + b_manchester, '--', label='Linear Regression Manchester', color='orange')
plt.fill_between(years, manchester_housing_pred_low, manchester_housing_pred_high, color='orange', alpha=0.2)
plt.xlabel('Year')
plt.ylabel('Total Housing Units')
plt.grid()
plt.legend(loc='upper left')
plt.title('Total Housing Units in Brighton & Hove and Manchester')
plt.show()

# %%
import pmdarima as pm
from sklearn.model_selection import train_test_split

train_x, test_x, train_y, test_y = train_test_split(bh_housing['Year'], bh_housing['Total housing units'], test_size=0.2, shuffle=False)

# Fit your model
model = pm.auto_arima(train_y, seasonal=True, m=12)

forecasts = model.predict(test_y.shape[0])  # predict N steps into the future

# Visualize the forecasts (blue=train, green=forecasts)
plt.figure(figsize=(20, 10))
```

```
plt.plot(train_x, train_y, c='blue')
plt.plot(test_x, test_y, c='green')
plt.plot(test_x, forecasts, '--', c='green')
plt.legend(['Train', 'Test', 'Forecasts'], loc='upper left')
plt.grid()
plt.title('ARIMA Model for Brighton & Hove')
plt.xlabel('Year')
plt.ylabel('Total Housing Units')
plt.show()


# %%
train_x, test_x, train_y, test_y = train_test_split(manchester_housing['Year'], manchester_housing['Total housing units'], test_size=0.2, shuffle=False)

# Fit your model
model = pm.auto_arima(train_y, seasonal=False)

# make your forecasts
forecasts = model.predict(test_y.shape[0])  # predict N steps into the future

# Visualize the forecasts (blue=train, green=forecasts)
plt.figure(figsize=(20, 10))
plt.plot(train_x, train_y, c='blue')
plt.plot(test_x, test_y, c='green')
plt.plot(test_x, forecasts, '--', c='green')
plt.legend(['Train', 'Test', 'Forecasts'], loc='upper left')
plt.grid()
plt.title('ARIMA Model for Manchester')
plt.xlabel('Year')
plt.ylabel('Total Housing Units')
plt.show()

# %%
# Perform senstivity analysis for the linear regression model
# For a small change in the model, how much does the output change?
```

```
# Check for Brighton & Hove.
m_bh, b_bh = np.polyfit(bh_housing['Year'].dt.year, bh_housing['Total
    ↪ housing units'], 1)
default_pred = m_bh*years + b_bh
years = np.array([2031, 2041, 2071])
m = np.linspace(0.95*m_bh, 1.05*m_bh, 5)
b = np.linspace(0.95*b_bh, 1.05*b_bh, 5)

# Create a dataframe to store the results
sensitivity_bh_2031 = pd.DataFrame()
sensitivity_bh_2041 = pd.DataFrame()
sensitivity_bh_2071 = pd.DataFrame()
# Make the index the m and the columsn the b.
sensitivity_bh_2031.index = m
sensitivity_bh_2041.index = m
sensitivity_bh_2071.index = m

# Loop through the m and b values and calculate the predictions.
for b_val in b:
    sensitivity_bh_2031[b_val] = m*2031 + b_val
    sensitivity_bh_2041[b_val] = m*2041 + b_val
    sensitivity_bh_2071[b_val] = m*2071 + b_val

    # Calculate the percent difference from the default prediction.
    sensitivity_bh_2031[b_val] = (sensitivity_bh_2031[b_val] -
        ↪ default_pred[0])/default_pred[0]*100
    sensitivity_bh_2041[b_val] = (sensitivity_bh_2041[b_val] -
        ↪ default_pred[1])/default_pred[1]*100
    sensitivity_bh_2071[b_val] = (sensitivity_bh_2071[b_val] -
        ↪ default_pred[2])/default_pred[2]*100

# Save the results to a csv file.
sensitivity_bh_2031.to_csv('sensitivity_bh_2031.csv')
sensitivity_bh_2041.to_csv('sensitivity_bh_2041.csv')
sensitivity_bh_2071.to_csv('sensitivity_bh_2071.csv')
print(sensitivity_bh_2031)

# Check for Manchester
```

```python
m_manchester, b_manchester = np.polyfit(manchester_housing['Year'].dt.year, manchester_housing['Total housing units'], 1)
default_pred = m_manchester*years + b_manchester
years = np.array([2031, 2041, 2071])
m = np.linspace(0.95*m_manchester, 1.05*m_manchester, 5)
b = np.linspace(0.95*b_manchester, 1.05*b_manchester, 5)

# Create a dataframe to store the results
sensitivity_manchester_2031 = pd.DataFrame()
sensitivity_manchester_2041 = pd.DataFrame()
sensitivity_manchester_2071 = pd.DataFrame()

# Make the index the m and the columsn the b.
sensitivity_manchester_2031.index = m
sensitivity_manchester_2041.index = m
sensitivity_manchester_2071.index = m

# Loop through the m and b values and calculate the predictions.
for b_val in b:
    sensitivity_manchester_2031[b_val] = m*2031 + b_val
    sensitivity_manchester_2041[b_val] = m*2041 + b_val
    sensitivity_manchester_2071[b_val] = m*2071 + b_val

    # Calculate the percent difference from the default prediction.
    sensitivity_manchester_2031[b_val] = (sensitivity_manchester_2031[b_val] - default_pred[0])/default_pred[0]*100
    sensitivity_manchester_2041[b_val] = (sensitivity_manchester_2041[b_val] - default_pred[1])/default_pred[1]*100
    sensitivity_manchester_2071[b_val] = (sensitivity_manchester_2071[b_val] - default_pred[2])/default_pred[2]*100

# Save the results to a csv file.
sensitivity_manchester_2031.to_csv('sensitivity_manchester_2031.csv')
sensitivity_manchester_2041.to_csv('sensitivity_manchester_2041.csv')
sensitivity_manchester_2071.to_csv('sensitivity_manchester_2071.csv')
```

```python

print(sensitivity_manchester_2031)


```

A.2 Python Code for Question 2

```python
# %%
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
import statsmodels.api as sm

# Load the data
# Go one folder back
os.chdir("..")

bh_path =
↪ "C:\\Users\\sapta\\OneDrive\\Documents\\GitHub\\M3Challenge2024\\data\\bh_homele
bh_hl = pd.read_csv(bh_path)

# Convert the Year column to datetime
bh_hl['Year'] = pd.to_datetime(bh_hl['Year'], format='%Y')

# Convert all other columns to numeric.
for col in bh_hl.columns[1:]:
    if bh_hl[col].dtype == 'object':
        # If data contains '-', use linear interpolation to fill
        ↪ missing values.
        if ' -' in bh_hl[col].unique():
            bh_hl[col] = bh_hl[col].replace(' -', np.nan)

        # Remove commas and convert to numeric
        bh_hl[col] = pd.to_numeric(bh_hl[col].str.replace(',', ''))

        bh_hl[col] = bh_hl[col].interpolate(method='linear')

```

```python
print(bh_hl)


hl_path = "C:\\Users\\sapta\\OneDrive\\Documents\\GitHub\\M3Challenge2024\\data\\mancheste
manchester_hl = pd.read_csv(hl_path)

# Convert the Year column to datetime
manchester_hl['Year'] = pd.to_datetime(manchester_hl['Year'], format='%Y')

# Convert all other columns to numeric.
for col in manchester_hl.columns[1:]:
    if manchester_hl[col].dtype == 'object':
        # If data contains '-', use linear interpolation to fill missing values.
        if ' -' in manchester_hl[col].unique():
            manchester_hl[col] = manchester_hl[col].replace(' -', np.nan)

        # Remove commas and convert to numeric
        manchester_hl[col] = pd.to_numeric(manchester_hl[col].str.replace(',', ''))

        manchester_hl[col] = manchester_hl[col].interpolate(method='linear')

print(manchester_hl)

# %%
bh_hl['total homeless'] = bh_hl['Homeless with priority need'] + bh_hl['Homeless without priority need']
bh_hl['homeless percent'] = bh_hl['total homeless'] / bh_hl['Total number of households']

manchester_hl['total homeless'] = manchester_hl['Homeless with priority need'] + manchester_hl['Homeless without priority need']
manchester_hl['homeless percent'] = manchester_hl['total homeless'] / manchester_hl['Total number of households']
```

```
# Plot the homeless percent
plt.figure(figsize=(10, 6))
plt.plot(bh_hl['Year'], bh_hl['homeless percent'], label='Brighton &
↪ Hove')
plt.plot(manchester_hl['Year'], manchester_hl['homeless percent'],
↪ label='Manchester')
plt.xlabel('Year')
plt.ylabel('Homeless Percent')
plt.title('Homeless Percent in Brighton & Hove and Manchester')
plt.grid()
plt.legend()
plt.show()


# %%
# Fit Manchester data to a sine and logistic function
def sine(x, a, b, c, d):
    return a * np.sin(b * x + c) + d

def logistic(x, a, b, c, d):
    return a / (1 + np.exp(-b * (x - c))) + d

def line(x, a, b):
    return a * x + b

def logistic_sine(x, a, b, c, d, e, f, g):
    return a / (1 + np.exp(-b * (x - c))) + d + e * np.sin(f * x + g)

def line_sine(x, a, b, c, d, e, f):
    return a * x + b + c * np.sin(d * x + e) + f

# %%
from scipy.optimize import curve_fit
from scipy.stats.distributions import t

popt1, pcov1 = curve_fit(sine, bh_hl['Year'].dt.year, bh_hl['homeless
↪ percent'], p0=[0.1, 0.1, 0.1, 0.1])

```

```
# Fit the logistic function
plt.figure(figsize=(10, 6))
plt.plot(bh_hl['Year'].dt.year, bh_hl['homeless percent'],
    label='Brighton & Hove')
plt.plot(bh_hl['Year'].dt.year, sine(bh_hl['Year'].dt.year, *popt1),
    label='Sine Fit')
plt.xlabel('Year')
plt.ylabel('Homeless Percent')
plt.title('Homeless Percent in Brighton & Hove')
plt.grid()
plt.legend()
plt.show()

# %%
# Predict for the next 50 years.
time = np.arange(2008, 2072, 1)
pred = sine(time, *popt1)

plt.figure(figsize=(20, 10))
plt.plot(bh_hl['Year'].dt.year, bh_hl['homeless percent'],
    label='Brighton & Hove')
plt.plot(time, pred)
plt.xlabel('Year')
plt.ylabel('Homeless Percent')
plt.title('Homeless Percent in Brighton & Hove')
plt.grid()
plt.legend(['Actual', 'Predicted'], loc='upper left')
plt.axvline(x=2031, color='r', linestyle='--')
plt.axvline(x=2041, color='r', linestyle='--')
plt.axvline(x=2071, color='r', linestyle='--')

dof = np.size(bh_hl['Year'].dt.year) - 1 # degrees of freedom:
# calculate student-t value
a = 0.05 #(1-0.95, 95% CI)
tval = t.ppf(1.0-a/2, dof)

ci = tval*np.sqrt(pcov1)
for i in range(len(popt1)):
    print("p{0}: {1} +/- {2}".format(i, popt1[i], ci[i, i]))
```

```
print('\n2031', pred[2031 - 2008]*100)
print('2041', pred[2041 - 2008]*100)
print('2071', pred[2071 - 2008]*100)

plt.show()

# %%
# Fit the data to logistic_sine using scipy

popt2, pcov2 = curve_fit(logistic_sine, manchester_hl['Year'].dt.year, manchester_hl['homeless percent'],
                          p0=[0.1, 0.5, 2000, 40, 0.1, 1, 1])

popt3, pcov3 = curve_fit(line_sine, manchester_hl['Year'].dt.year, manchester_hl['homeless percent'],
                          p0=[0.1, 0.5, 4000, 30, 0.1, 1])

# Plot the data and the fitted curve
plt.figure(figsize=(10, 6))
plt.plot(manchester_hl['Year'].dt.year, manchester_hl['homeless percent'], label='Manchester')
# plt.plot(manchester_hl['Year'], logistic_sine(manchester_hl['Year'].dt.year, *popt), label='Logistic Sine Fit')
plt.plot(manchester_hl['Year'].dt.year, line_sine(manchester_hl['Year'].dt.year, *popt3), label='Logistic Fit')
plt.xlabel('Year')
plt.ylabel('Homeless Percent')
plt.title('Homeless Percent in Manchester')
plt.grid()
plt.legend()
plt.show()

# %%
# Predict for the next 50 years.
time = np.arange(2008, 2072, 1)
pred = line_sine(time, *popt3)
```

```
plt.figure(figsize=(20, 10))
plt.plot(manchester_hl['Year'].dt.year, manchester_hl['homeless
  ↪ percent'], label='Manchester')
plt.plot(time, pred)
plt.xlabel('Year')
plt.ylabel('Homeless Percent')
plt.title('Predicted Homeless Percent in Manchester')
plt.grid()
plt.legend(['Actual', 'Predicted'], loc='upper left')
plt.axvline(x=2031, color='r', linestyle='--')
plt.axvline(x=2041, color='r', linestyle='--')
plt.axvline(x=2071, color='r', linestyle='--')

print('2031', pred[2031 - 2008]*100)
print('2041', pred[2041 - 2008]*100)
print('2071', pred[2071 - 2008]*100)

plt.show()

# %%
from statsmodels.tsa.stattools import adfuller
test_result=adfuller(manchester_hl['homeless percent'])
test_result

# %%
manchester_hl['Seasonal first difference'] = manchester_hl['total
  ↪ homeless'] - manchester_hl['total homeless'].shift(6)
manchester_hl.head(14)

# %%
adfuller(manchester_hl['Seasonal first difference'].dropna())

# %%
from statsmodels.tsa.seasonal import seasonal_decompose
result = seasonal_decompose(manchester_hl['Year'])
result.plot()

# %%
```

```
from sklearn.svm import SVR
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

X_train, X_test, y_train, y_test = train_test_split(manchester_hl['Year'], manchester_hl['homeless percent'], train_size=0.8, test_size=0.2, shuffle=False)

X_train, X_test, y_train, y_test = X_train.to_frame(), X_test.to_frame(), y_train.to_frame(), y_test.to_frame()

regr = make_pipeline(StandardScaler(), SVR(C=0.5, epsilon=5))

regr.fit(X_train, y_train)
forecasts = regr.predict(X_test)

# x = np.arange(X_train + X_test)
plt.plot(X_train, y_train, c='blue')
plt.plot(X_test, y_test, c='red')
plt.plot(X_test, forecasts, c='green')
plt.show()

# %%
import pmdarima as pm
from sklearn.model_selection import train_test_split

manchester_hl['total homeless'] = manchester_hl['Homeless with priority need'] + manchester_hl['Homeless without priority need']

manchester_hl['homeless percent'] = manchester_hl['total homeless'] / manchester_hl['Total number of households']

X_train, X_test, y_train, y_test = train_test_split(manchester_hl['Year'], manchester_hl['total homeless'], train_size=0.7, shuffle=False)

model = pm.auto_arima(y_train, seasonal=True, m=3, seasonal_test='ch')

```

```
forecasts = model.predict(X_test.shape[0])

# x = np.arange(X_train + X_test)
plt.plot(X_train, y_train, c='blue')
plt.plot(X_test, y_test, c='red')
plt.plot(X_test, forecasts, c='green')
plt.show()

# %%


# %%



```

A.3 Python Code for Question 3

```
# %%
# Make a ML model to predict the target variable
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os

# Load the data
# Go one folder back
os.chdir("..")

path =
↪ "C:\\Users\\sapta\\OneDrive\\Documents\\GitHub\\M3Challenge2024\\data\\homelessn
data = pd.read_csv(path)

# Convert the Year column to datetime
data['Year'] = pd.to_datetime(data['Year'], format='%Y')

# Convert all other columns to numeric.
for col in data.columns[1:]:
```

```
    if data[col].dtype == 'object':
        # If data contains '-', use linear interpolation to fill missing values.
        if ' -' in data[col].unique():
            data[col] = data[col].replace(' -', np.nan)

        # Remove commas and convert to numeric
        data[col] = pd.to_numeric(data[col].str.replace(',', ''))

        data[col] = data[col].interpolate(method='linear')

print(data)

# %%
data['Total Homeless'] = data['Homeless, priority need'] + data['Homeless, not priority need']
data['Homeless Percent'] = data['Total Homeless'] / data['Total number of households']
data['Housing Disparity'] = data['Total number of households'] - data['Total Housing Units']
# data['Population Change'] = data['Total Population'] - data['Total Population'].shift(1)
# data['Homeless Change'] = data['Total Homeless'] - data['Total Homeless'].shift(1)

data.drop(['Homeless, priority need', 'Homeless, not priority need', 'Eligible but not homeless'], axis=1, inplace=True)

# Make a correlation matrix
correlation_matrix = data.corr()
sns.heatmap(correlation_matrix, annot=True)
plt.title('Correlation Heat Map')
# data.drop(['Homeless Percent'], axis=1, inplace=True)
plt.show()

data_copy = data.copy()

# %%
from sklearn.model_selection import train_test_split
```

```python

# Convert the Year column to numeric
data['Year'] = data['Year'].dt.year

# Split the data into training and testing sets
# Drop other variable
data.fillna(0, inplace=True)

data.drop(['Homeless Percent'], axis=1, inplace=True)


X = data.drop(['Total Homeless'], axis=1)
y = data['Total Homeless']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
    random_state=42)

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Train the model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Calculate the mean squared error
mse = mean_squared_error(y_test, y_pred)
print('Mean Squared Error:', mse)

# R^2 score
r2 = model.score(X, y)
print('R^2:', r2)

# Print the coefficients
print('Coefficients:', model.coef_)

# Plot the fit.
```

```python
# X = np.linspace(data['Year'].min(), data['Year'].max() + 50, 1)
pred = model.predict(X)
plt.figure(figsize=(20, 10))
plt.scatter(data['Year'], data['Total Homeless'])
plt.plot(data['Year'], pred, color='red')
plt.title('Total Homeless Population vs Year for Brighton & Hove')
plt.xlabel('Year')
plt.ylabel('Total Homeless Population')
plt.legend(['Data', 'Model Fit'])
plt.grid()
plt.show()

# %%
# Make models for Total number of households, Total Population, Total
↪ Housing Units, and Median Income
total_households = data['Total number of households']
total_population = data['Total Population']
total_housing_units = data['Total Housing Units']
median_income = data['Median Income']

# Plot over time.
plt.plot(data['Year'], total_population)
plt.title('Total number of households vs Year')
plt.xlabel('Year')
plt.ylabel('Total number of households')
plt.show()

# %%
# Train a model
from pycaret.time_series import TSForecastingExperiment
s = TSForecastingExperiment()

s.setup(data=data_copy, session_id=42, target='Homeless Percent',
↪ numeric_imputation_target='mean',
↪ numeric_imputation_exogenous='mean')

best = s.compare_models()

# %%
```

```
# OOP API
s.plot_model(best, plot = 'forecast', data_kwargs = {'fh' : 50})
```