MathWorks Math Modeling Challenge
A program of SIAM

PREVIEW PAPER: ABOVE AVERAGE

The team's summary is addressed to the Secretary of the U.S. Department of Housing and Urban Development. Most of the papers examined in the pre-triage rounds did not feature a letter addressed to a government official, which is presented as an optional way to frame the executive summary. The team did a good job of stating an overview of the problem, discussed their methodology, and they explicitly stated most of their primary results.

Prior to addressing the first question, the team gave a good overview of the data that was provided to them but did not cite the source of the data. Their graphs have captions and are well annotated. The assumptions are clearly stated and include some justifications. It is noteworthy that the team provides both citations and references.

For the first question, the team made use of a time-series approach. The methodology used was more sophisticated than what most other teams employed. The team did a good job of stating their results and included a 95% confidence interval for the mean of their approximation. It is not clear, though, how the confidence interval was estimated. Unfortunately, the team did not go beyond stating a prediction and did not use their model to provide insight into the phenomena of interest nor provide insight into the relationship between the variables
of interest. The team's final predictions include the addition of an additive noise term, but its inclusion is poorly motivated.

For the second question, the team made use of a Grated Recurrence Units neural network. The team did a good job of justifying its use as well as describing the parameters of interest. They also clearly stated the methodology in how they used the data for training. Just as they did in the first question, the results were clearly stated. Again, the model was not used to provide insight into the problem itself.

The team's approach to the third question closely mirrored their approach for the second question. The results of their approximations were stated. The presentation was more terse than for the second question, and it was more difficult to interpret the results and put them in a broader context.

The team did provide a final conclusion. Their conclusion is relatively terse and restated their results. The conclusion was not used to provide additional insight or interpretation of the primary results.

Overall, this entry did not receive uniform scores. Some judges felt that the overall discussion and presentation as well as relatively good techniques were noteworthy. For other judges the lack of insights about the relationships between variables, the minimal interpretation of the results, as well as the use of an over-complicated method were problematic.

A Tale of Two Crises

Team 17970

March 2024

Executive Summary

To the Secretary of the U.S. Department of Housing and Urban Development:

Income inequality in the United States of America has been steadily increasing for a variety of reasons. One economic signal that demonstrates this trend is the steadily increasing Gini Coefficient, a measure of statistical dispersion intended to represent the income or wealth distribution of a nation's residents (In the past thirty years, the Gini Coefficient has grown from 0.43 to 0.47) [1]. As income inequality continues to increase over the coming years [2], presumably decades, conditions point to the worsening of the housing market [3]. We aim to show evidence-backed predictive models that demonstrate these trends, as well as offer a possible plan to ameliorate these crises.

First, we predict future vacant unit amounts across Seattle, Washington, and Albuquerque, New Mexico by using Holt-Winters, a time series forecasting model, specifically designed for capturing and predicting patterns in data that exhibit specific trends [4]. We use this model on data regarding vacant units from across the United States obtained from the Federal Reserve Bank of St. Louis [5]. After this, we compare this data to the data for specifically Albuquerque and Seattle, comparing them through adjustable multipliers. We then predict the trend of these multipliers using Auto-Regressive Integrated Moving Average (ARIMA). Finally, we use the ideas of random walk and Brownian motion to increase the model's reflection of the true housing market [6]. Using this model, we predict that the number of vacant units in Albuquerque would be 29319 in 2034, 23518 in 2044, and 9644 in 2074; in Seattle, it would be 25704 in 2034, 20515 in 2044, and 8868 in 2074.

Next, using GRU (Gated Recurrent Unit), a neural networking model involved in time series forecasting we extrapolated the pattern and applied it to the proximal future. We found that the housing supply generally increased over 50 years, finding a final homeless population of 2107 in 2074, an unusually high amount in today's day.

Thereafter, we used the same model again, GRU, to predict the homeless population after the solutions were implemented. We explained the proportional relationships between both variables and the homeless population rise. To do this, we varied the data by 10% in both variables, respectively, a 0.9 and 1.1 factor. Evaluation of these two variations in these variables yielded an appropriate solution to the problem at hand.

We believe that through these results, policy-makers and leaders in major industries will be able to gain insight on how to assist the growth of the economy and prevent major damage to the housing market.

Contents

1 It Was the Best of Times

1.1 Defining the Problem

The first problem tasks us with building a predictive model for the changes in housing supply for **Seattle, Washington** and **Albuquerque, New Mexico** over the next 10, 20, and 50 years.

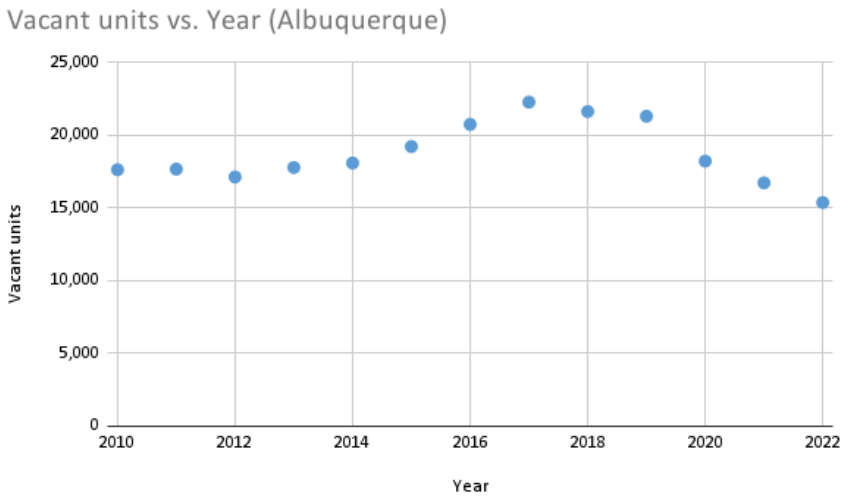

Figure 1: Sample Data Dot plot of Vacant Units over time (annually) in Albuquerque

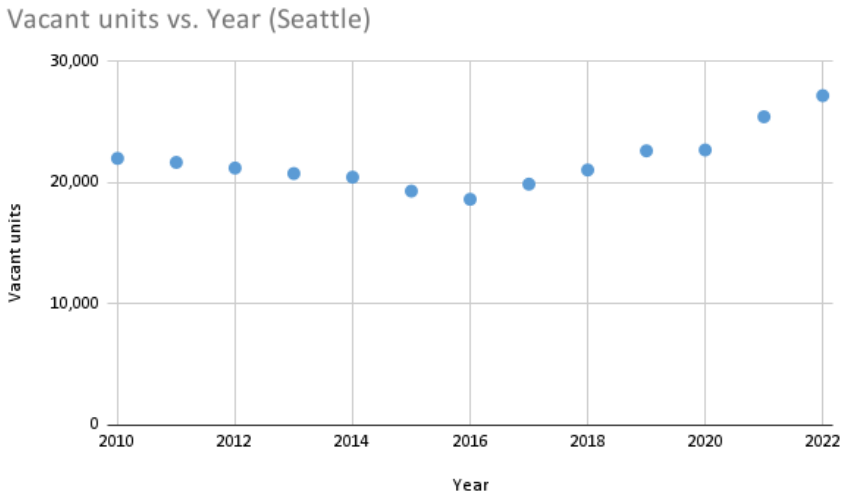

Figure 2: Sample Data Dot plot of Vacant Units over time (annually) in Seattle

Here we display our sample data for 2010 to 2022 for vacant housing [8]. In addition, we utilize data from the St. Louis Federal Reserve Economic public database, which tracks quarterly trends of vacant housing across the United States from 2000 to 2023 in thousands of units.

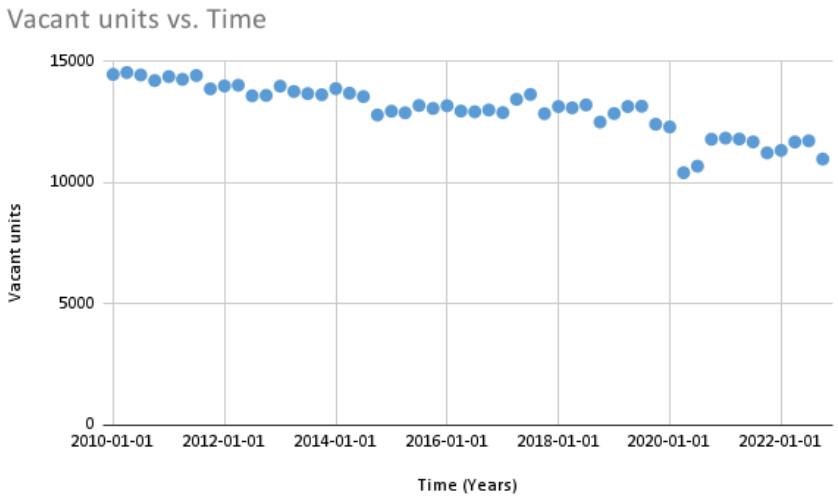

Figure 3: US Vacancy Data 2010-2022

1.2 Assumptions

The pattern of changes in the housing supply is related to the changes in the number of vacant units.

The number of vacant units reflects the supply and demand in the housing market. If supply is high, vacancy will be higher. On the other hand, if demand is high, vacancy will be lower [7].

The changes in the number of vacant units in Albuquerque and Seattle are related by adjustable moving multipliers to the number of vacant units in the United States as a whole.

Because most housing developments are reflected in urban areas [9], and Seattle and Albuquerque are both major cities, it's fair to assume that changes in the number of vacant units in these two cities will be closely related to changes in the number of vacant units in the entire country.

We can limit the data we consider to the period after 2010.

The housing crisis which began in the year 2007, and was a part of the larger global financial crisis, greatly changed the housing market [10]. Thus, any data before the year the housing crisis ended, which was 2010 [10], would likely be irrelevant to the modern market.

The housing market has some characteristics of Brownian motion.

The factors influencing the housing market are largely chaotic and unpredictable, so it's fair to add a random walk as some noise to a larger general trend that we develop through some other regression method [6]. This has precedent in other similar financial systems such as the stock market, where random walks were at times used to make predictions due to the chaos in the system [11]. All subsequent models are trained on noisy models, allowing them to effectively adapt to chaos and identify key trends in the market.

1.3 Variables Considered

On review of the significant values and trends, we chose to omit sections of the provided data on a multitude of factors including specific numbers of clients, Boats, RVs, Vans, and other forms of mobile housing.

Symbol	Definition	Units
t	Time	Year
$V_A(t)$	Number of vacant houses in Albuquerque for a given year	Houses
$V_S(t)$	Number of vacant houses in Seattle for a given year	Houses
$V_U(t)$	Number of vacant houses in the United States for a given year	Houses
$M_A(t)$	The ratio of the number of vacant houses in Albuquerque to the number of vacant houses in the United States (adjustable multiplier)	None
$M_S(t)$	The ratio of the number of vacant houses in Seattle to the Number of vacant houses in the United States (adjustable multiplier)	None
$\hat{V}_U(t)$	Predicted value of $V_U(t)$	Houses
$\hat{M}_A(t)$	Predicted value of $M_A(t)$	Houses
$\hat{M}_S(t)$	Predicted value of $V_S(t)$	Houses
$\hat{V}_A(t)$	Predicted value of $V_A(t)$	Houses
$\hat{V}_S(t)$	Predicted value of $V_S(t)$	Houses

Figure 4

Note that

$$M_A(t) = \frac{V_A(t)}{V_U(t)}$$

and

$$M_S(t) = \frac{V_S(t)}{V_U(t)}.$$

1.4 The Model

Developing the Model

All computational tasks were conducted using Python in the Google Colaboratory Cloud IDE. Packages used included *Pandas*, *Numpy*, *TensorFlow*, and *statsmodel*.

Because the housing market has a plethora of complex factors, we do not see any use in attempting to model $\hat{V}_A(t)$ or $\hat{V}_S(t)$ by linear regression, which would be too simplistic for the sake of our model. While linear regression could be effective for predicting general trends, it is unlikely that the specific number of vacant houses will either be strictly increasing or decreasing over time. A similar fact is true for a polynomial regression.

We elect to use Holt-Winters exponential smoothing regression from the *statsmodel* package to extrapolate future data for the $V_U(t)$. This is because the Holt-Winters method places relatively more weight on recent observations [4]. In the housing market, this characteristic is crucial because trends in the market depend highly on the current characteristics of the market [12]. These characteristics are volatile and chaotic, thus it is impractical to place an equal or higher weight on observations that are not recent. Furthermore, Holt-Winters regression predicts future trends relatively well, making it a good fit for extrapolating trends. There is also a precedent of Holt-Winters being used in economics and related fields, which is in the vein of what we are investigating [4]. Finally, the external data on vacant units across the United States was used as it has finer granularity than our provided data, enabling us to make more robust and accurate predictions on future trends based on the exponential smoothing regressor.

Executing the Model

We use the additive method since seasonal variations are approximately constant through the series [13].

Setting $\alpha \in [0, 1]$, $\beta \in [0, 1]$, and $\gamma \in [0, 1]$ as the data smoothing factor, trend smoothing factor, and seasonal change smoothing factor, respectively, and setting m to be the number of periods in a seasonal cycle, we use the

following

$$\begin{aligned}\hat{V}_U(t+h) &= \ell(t) + hb(t) + s(t-m+1),\\ \ell(t) &= \alpha(V_U(t) - s(t-m)) + (1-\alpha)(\ell(t-1) + b(t-1)),\\ b(t) &= \beta(\ell(t) - \ell(t-1)) + (1-\beta)\cdot b(t-1),\\ s(t) &= \gamma(V_U(t) - \ell(t-1) - b(t-1)) + (1-\gamma)\cdot s(t-m).\end{aligned}$$

Using existing data for $V_U(t)$, we aimed to predict $V_A(t)$ and $V_S(t)$ by plotting the adjustable multipliers $M_A(t)$ and $M_S(t)$. We have the following data:

t	$\frac{V_U(t)}{1000}$	$\frac{V_A(t)}{1000}$	$\frac{V_S(t)}{1000}$	$M_A(t)$	$M_S(t)$
2010	14471	22.012	17.635	0.0015	0.0012
2011	14382	21.684	17.675	0.0015	0.0012
2012	13995	21.218	17.134	0.0015	0.0012
2013	13985	20.766	17.786	0.0015	0.0013
2014	13885	20.464	18.093	0.0015	0.0013
2015	12952	19.317	19.228	0.0015	0.0015
2016	13179	18.638	20.750	0.0014	0.0016
2017	12895	19.889	22.283	0.0015	0.0017
2018	13146	21.057	21.634	0.0016	0.0016
2019	12858	22.639	21.310	0.0018	0.0017
2020	12304	22.708	18.225	0.0018	0.0015
2021	11844	25.448	16.733	0.0021	0.0014
2022	11337	27.190	15.378	0.0024	0.0014

Figure 5

We plot the data for $M_A(t)$ and $M_S(t)$ as follows:

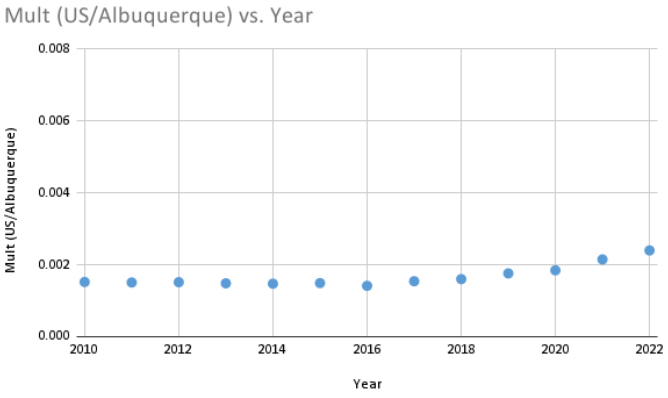

Figure 6: Multiplier Value (Albuquerque)

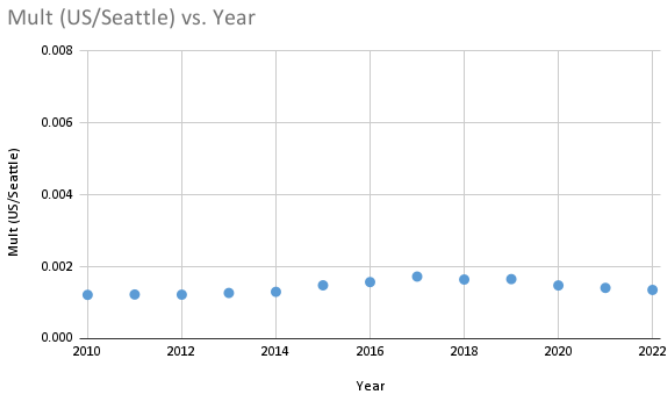

Figure 7: Multiplier Value (Seattle)

After obtaining this data, we look for trends in $M_A(t)$ and $M_S(t)$. We use ARIMA (Auto-Regressive Integrated Moving Average) from the *statsmodel* package, which is optimal for making predictions in the context of repeating short term patterns [14], and as seen in the graphs above, there are very slight fluctuations that imply the necessity of a short-term perspective.

We combine our projections for $\hat{V}_U(t)$ (from the Holt-Winters Exponential Regressor) with our projections for $\hat{M}_A(t)$ and $\hat{M}_S(t)$ as follows:

$$\hat{V}_A(t) = \hat{M}_A(t) \cdot \hat{V}_U(t),$$
$$\hat{V}_S(t) = \hat{M}_S(t) \cdot \hat{V}_U(t).$$

After obtaining the trend, we apply a random walk to generate a random sequence of terms that we added to the projection in each year. Recall that we want to apply a random walk because we assumed that the housing market has characteristics comparable to Brownian motion. The terms of the random walk are generated by a Gaussian distribution, specifically with a mean of 0 and a standard deviation of 300. We chose this standard deviation based on the average of the value $|V_A(t) - V_A(t-1)|$, and the analogous value for $V_S(t)$, based on our existing data.

1.5 Results

Through Holt-Winters' execution, we determine the forecasting vacancy values for 2034, 2044, and 2074 to be 29319, 23518, and 9644 respectively. Their respective, 95% confidence intervals were discovered through the use of ARIMA[1] (Auto-Regressive Integrated Moving Average), due to its application with the multipliers*.

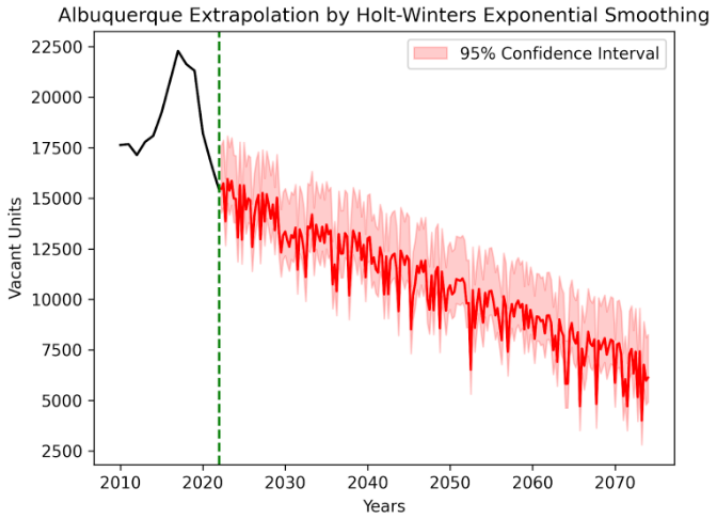

Figure 8: Holt-Winters' Albuquerque Extrapolation

[1] *Explained in* ***Justification***

Years after 2024	t	$\hat{V}_A(t)$	**Confidence Interval**
10	2034	29319	$(27342, 30557)$
20	2044	23518	$(21541, 24756)$
50	2074	9644	$(8406, 11621)$

Figure 9

Here, we display the results of our Holt-Winters extrapolation as well as our 95% confidence intervals for Albuquerque. The values for the years decided are isolated in the chart below.

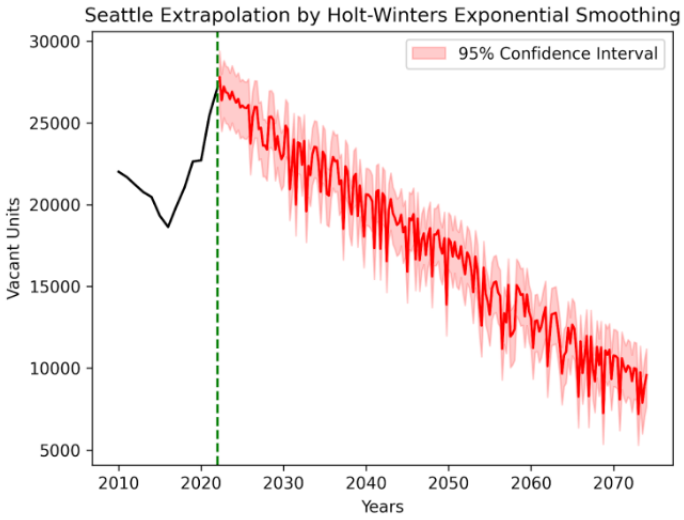

Figure 10: Holt Winters' Seattle Extrapolation

Years after 2024	t	$\hat{V}_S(t)$	**Confidence Interval**
10	2034	25704	$(23804, 27301)$
20	2044	20515	$(18615, 22112)$
50	2074	8868	$(6968, 10465)$

Figure 11

Again, we display the same values, now, for Seattle.

Based on these trends, we forecast significant decreases in the expected number of vacant units in both Albuquerque and Seattle, with an expected 35.7% and 67.8% reduction from 2024 to 2074, respectively.

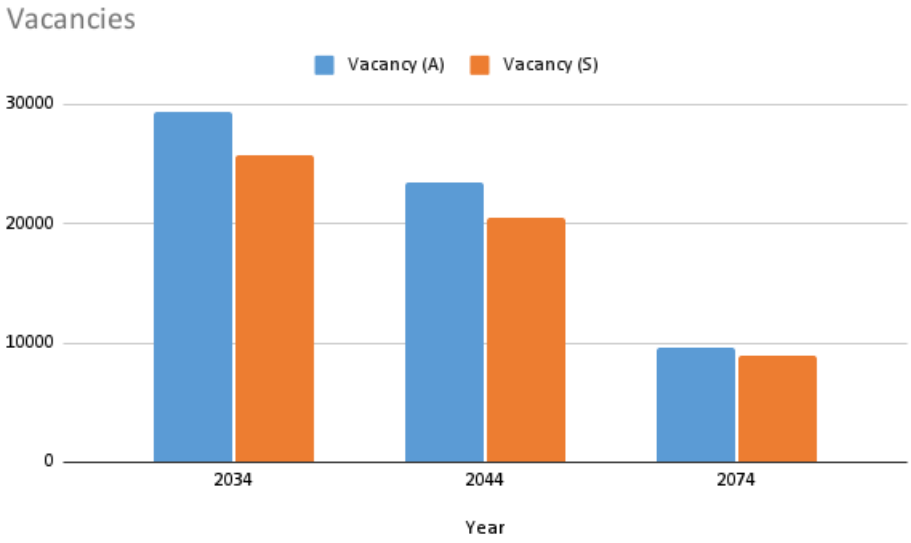

Figure 12: Total Vacancies for Q1 regarding Albuquerque (A) and Seattle (S)

1.6 Justification

Initially, we were met at a crossroads with the choice between fitting with Holt-Winters Exponential Smoothing and ARIMA, both apt time series forecasting methods commonly used in statistics, econometrics, and here extrapolation [4] [14].

Holt-Winters is particularly well-suited for time series data, with additional benefits with seasonal data [4]. Additionally, Holt-Winters is known to be better for long-term projection whereas ARIMA prevails with short-term [4] [14]. Here, our objective was to derive values for 10, 20, and 50 years, so Holt-Winters was selected (Seasonal granulation would be impractical given these **yearly** predictions, so we forwent its usage).

Explanation of the Model: Holt-Winters decomposes the time series into component values then uses these to forecast future plots.

Holt-Winters uses exponential smoothing to handle trends in the data. This makes it more adaptive to changes in the underlying trend over time [4]. ARIMA, while capable of capturing trends, might not be as flexible in adjusting to changing trends as Holt-Winters [14]. As we recognized the underlying trends and changes in these trends in our sample data and how it was applicable to the extrapolation, the data was fit with Holt-Winters. Attempting this problem we chose to employ the use of US-wide vacancy data. Finding a direct relationship with the trends in both Albuquerque and Seattle, we chose to utilize ARIMA to forecast multiplying factors.

It is important to note a particular weakness in our model, which is the assumption about the multiplier between vacant units in the US and in Albuquerque and Seattle. Even if this assumption is a solid one to make, it may not be fully accurate and has a drastic impact on our model. Furthermore, our model is based on data from very recent years, which may have been greatly impacted by the COVID-19 pandemic [15]. This presents another weakness in our projections.

Another weakness sprouts from our model of univariate time series forecast. Regarding our model, Holt-Winters does benefit from a seasonal distribution as granularity does assist its fit. The problem at hand does not regard a seasonal nor monthly prediction so we opted to use data at an annual level and not at a more specific extent. This may have somewhat hindered our predictive values.

2 It Was the Worst of Times

2.1 Defining the Problem

The second problem tasks us with building a predictive model for the changes in the homeless population for **Seattle, Washington** and **Albuquerque, New Mexico** over the next 10, 20, and 50 years. The figures below are from the provided data [8].

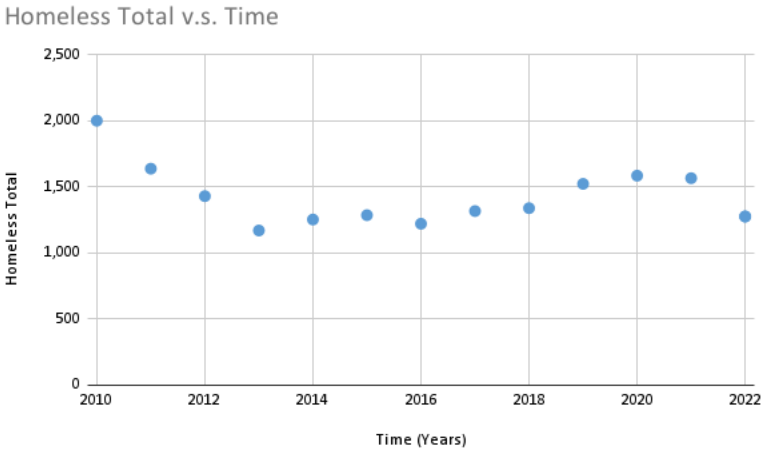

Figure 13: Albuquerque Homeless Population over Time

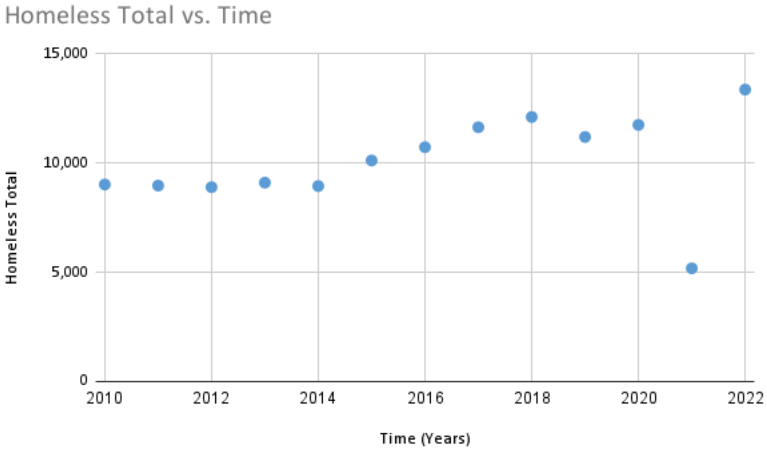

Figure 14: Seattle Homeless Population over Time

2.2 Assumptions

The number of homeless people in a given area is closely related to the number of vacant units.

In general, a lower number of vacant units would mean the housing market is more competitive, so there would be more people unable to purchase a house [16].

The number of homeless people also exhibits characteristics of Brownian motion.

We assumed that the vacant units had characteristics of Brownian motion [6], and also that the number of homeless people is closely related to the number of vacant units [16]. Thus, the number of homeless people should also have characteristics of Brownian motion, and we can apply the properties of random walk on our model.

2.3 Variables Considered

As will be explained further, we apply a deep learning model to make projections into the future. As such, we consider all the variables that were provided, albeit with varying weights.

Symbol	Definition	Units
t	Time	Year
$V_A(t)$	Number of vacant houses in Albuquerque for a given year	Houses
$V_S(t)$	Number of vacant houses in Seattle for a given year	Houses
$P_A(t)$	Population of Albuquerque for a given year	People
$P_S(t)$	Population of Seattle for a given year	People
$A_A(t)$	Median age of Albuquerque for a given year	Year
$A_S(t)$	Median age of Seattle for a given year	Year
$E_A(t)$	Emergency sheltered population of Albuquerque for a given year	People
$E_S(t)$	Emergency sheltered population of Seattle for a given year	People
$T_A(t)$	Transitional housing population of Albuquerque for a given year	People
$T_S(t)$	Transitional housing population of Seattle for a given year	People
$S_A(t)$	Total sheltered population of Albuquerque for a given year	People
$S_S(t)$	Total sheltered population of Seattle for a given year	People
$U_A(t)$	Total unsheltered population of Albuquerque for a given year	People
$U_S(t)$	Total unsheltered population of Seattle for a given year	People
$H_A(t)$	Total homeless population of Albuquerque for a given year	People
$H_S(t)$	Total homeless population of Seattle for a given year	People
$I_A(t)$	Median household income of Albuquerque for a given year (inflation-adjusted for US dollars)	USD
$I_S(t)$	Median household income of Seattle for a given year (inflation-adjusted for US dollars)	USD

Figure 15

We also define the following proportions as variables included in our calculations:

Symbol	Definition
$p_{1,A}(t)$	Proportion of population in Albuquerque with income less than $\$10,000$
$p_{1,S}(t)$	Proportion of population in Seattle with income less than $\$10,000$
$p_{2,A}(t)$	Proportion of population in Albuquerque with income between $\$10,000$ and $\$14,999$
$p_{2,S}(t)$	Proportion of population in Seattle with income between $\$10,000$ and $\$14,999$
$p_{3,A}(t)$	Proportion of population in Albuquerque with income between $\$15,000$ and $\$24,999$
$p_{3,S}(t)$	Proportion of population in Seattle with income between $\$15,000$ and $\$24,999$
$p_{4,A}(t)$	Proportion of population in Albuquerque with income between $\$25,000$ and $\$34,999$
$p_{4,S}(t)$	Proportion of population in Seattle with income between $\$25,000$ and $\$34,999$
$p_{5,A}(t)$	Proportion of population in Albuquerque with income between $\$35,000$ and $\$49,999$
$p_{5,S}(t)$	Proportion of population in Seattle with income between $\$35,000$ and $\$49,999$
$p_{6,A}(t)$	Proportion of population in Albuquerque with income between $\$50,000$ and $\$74,999$
$p_{6,S}(t)$	Proportion of population in Seattle with income between $\$50,000$ and $\$74,999$
$p_{7,A}(t)$	Proportion of population in Albuquerque with income between $\$75,000$ and $\$99,999$
$p_{7,S}(t)$	Proportion of population in Seattle with income between $\$75,000$ and $\$99,999$
$p_{8,A}(t)$	Proportion of population in Albuquerque with income between $\$100,000$ and $\$149,999$
$p_{8,S}(t)$	Proportion of population in Seattle with income between $\$100,000$ and $\$149,999$
$p_{9,A}(t)$	Proportion of population in Albuquerque with income between $\$150,000$ and $\$199,999$
$p_{9,S}(t)$	Proportion of population in Seattle with income between $\$150,000$ and $\$199,999$
$p_{10,A}(t)$	Proportion of population in Albuquerque with income greater than $\$200,000$
$p_{10,S}(t)$	Proportion of population in Seattle with income greater than $\$200,000$
$p_{P,A}(t)$	Proportion of population in Albuquerque at or below poverty level
$p_{P,S}(t)$	Proportion of population in Seattle at or below poverty level

Figure 16

Note that

$$\sum_{k=1}^{10} p_{k,A}(t) = \sum_{k=1}^{10} p_{k,S}(t) = 1.$$

2.4 The Model

Developing the Model

While our previous univariate exponential smoothing method may enable long-term forecasting, in order to model trends of homelessness, we implement a multivariate method which could better grasp the interconnected nature of our variables. To better understand these causal networks between our 13 variables, we apply the Granger Causality Test, which allows us to identify the effect one time series may have on another. Based on the significance test, which we highlight in Section 2.5, we find these variables are heavily interconnected, showcasing the necessity for accurate multivariate modeling.

In order to accurately identify these temporal trends, we utilize a Gated Recurrent Units (GRU) neural network architecture using the *TensorFlow* package with 50 units and a fully interconnected Dense layers which outputs our 21 variables, with a total of 10,221 trainable parameters. This architecture is a type of recurrent neural network (RNN) architecture that has been designed to address some of the limitations of traditional RNNs, such as the vanishing gradient problem. While GRUs share similarities with the more commonly known Long Short-Term Memory (LSTM) networks, they have a slightly simpler structure with fewer parameters, better suited for our data.

To train our model, we split our given data for Seattle and Albuquerque with a 80:20 training and test split. We then train over 50 epochs, plotting training and validation root mean squared error (RMSE). Given the significant decrease in RMSE, we can determine the model converged to an optimal state without overfitting.

Executing the Model

We use a fully gated unit using the sigmoid and hyperbolic tangent activation functions [17],

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

and

$$\phi(x) = \tanh(x).$$

Define the binary operation $\odot$ between $m \times n$ real matrices, known as the Hadamard product [18], by

$$(A \odot B)_{ij} = A_{ij} \cdot B_{ij}.$$

Defining $x_{t,A}, x_{t,S} \in \mathbb{R}^{20}$ such that each component of each vector corresponds to a variable listed in Section 2.3 (we omit t since all other variables are dependent on t), we apply the GRU to compute the output vector, h_t, as follows:

$$\begin{aligned}
z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z), \\
r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r), \\
\hat{h}_t &= \phi(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h), \\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t.
\end{aligned}$$

2.5 Results

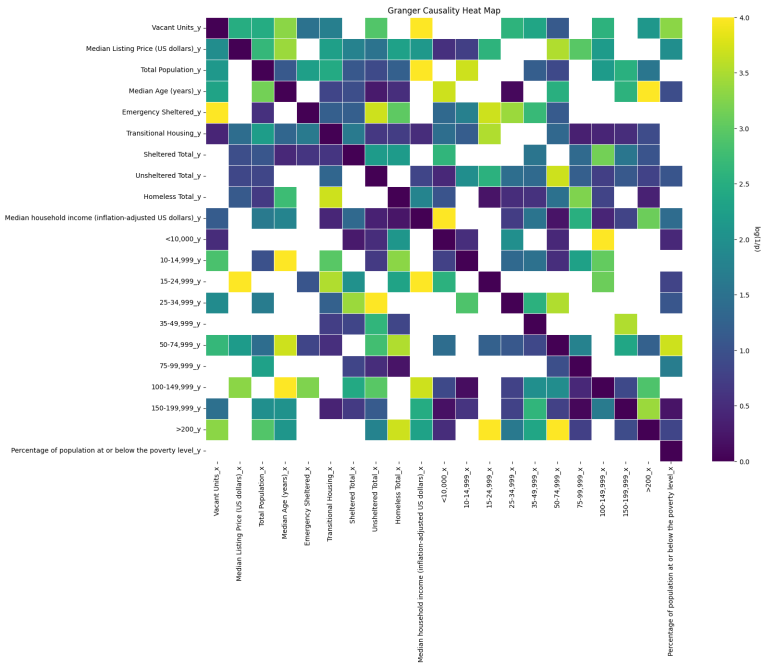

Figure 17: Granger Causality Test

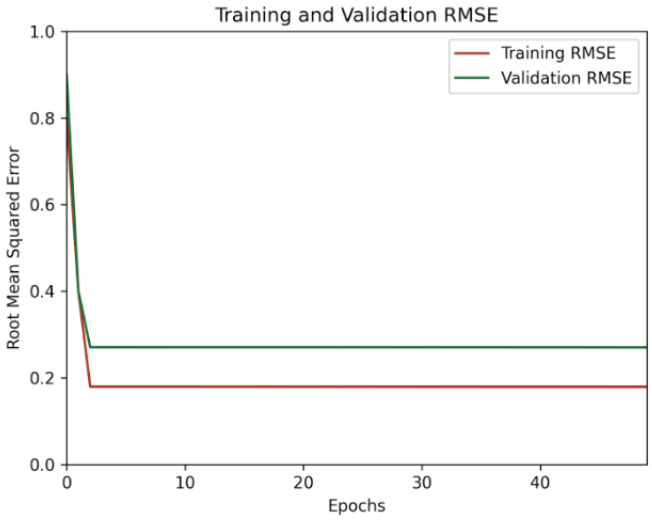

Figure 18: Training and Validation Loss

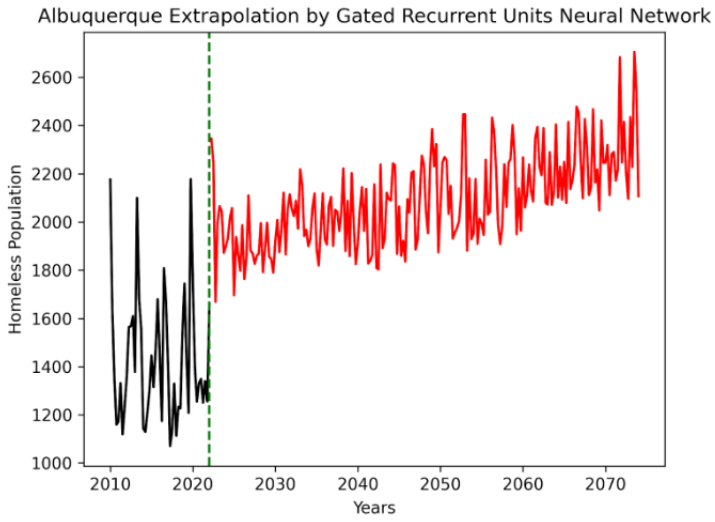

Figure 19: Albuquerque

Years after 2024	P_A
10	1668
20	1929
50	2107

Figure 20: Albuquerque

Here, we present the GRU given and forecasted data for the unhoused population in Albuquerque as well as the isolated yearly points below.

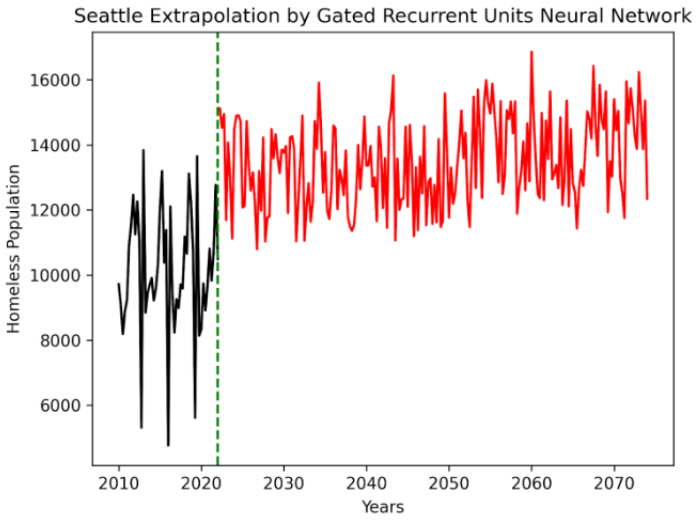

Figure 21: Seattle

Years after 2024	P_S
10	13982
20	14674
50	16343

Figure 22: Seattle

Again, we display the GRU forecasted data for homelessness, now for Seattle.

2.6 Reflection

Our GRU neural network predicts increases in the homeless population in both Albuquerque and Seattle, with a 31.7% and 19.3% increase respectively. These aligns with common trends seen in existing predictions, which identify the homeless crisis as continuing to increase at worsening rates.

3 Rising from This Abyss

3.1 Defining the Problem

The third problem tasks us with building a model that is both adaptive and predictive for creating a plan to lower homelessness in **Seattle, Washington** and **Albuquerque, New Mexico**.

3.2 Assumptions

The 10% respective decrease and increase in population and median household income directly influences the homeless

In general, given all the factors researched and identified, we cannot determine any other significant variables influencing the rise in the homeless population. We can only attribute these main two factors to the prominent issue.[2]

3.3 Variables Considered

Similarly to in Question 2, we apply a deep learning model to make projections into the future. Again, we consider all the variables that were provided, albeit with varying weights.

Symbol	Definition	Units
t	Time	Year
$V_A(t)$	Number of vacant houses in Albuquerque for a given year	Houses
$V_S(t)$	Number of vacant houses in Seattle for a given year	Houses
$P_A(t)$	Population of Albuquerque for a given year	People
$P_S(t)$	Population of Seattle for a given year	People
$A_A(t)$	Median age of Albuquerque for a given year	Year
$A_S(t)$	Median age of Seattle for a given year	Year
$E_A(t)$	Emergency sheltered population of Albuquerque for a given year	People
$E_S(t)$	Emergency sheltered population of Seattle for a given year	People
$T_A(t)$	Transitional housing population of Albuquerque for a given year	People
$T_S(t)$	Transitional housing population of Seattle for a given year	People
$S_A(t)$	Total sheltered population of Albuquerque for a given year	People
$S_S(t)$	Total sheltered population of Seattle for a given year	People
$U_A(t)$	Total unsheltered population of Albuquerque for a given year	People
$U_S(t)$	Total unsheltered population of Seattle for a given year	People
$H_A(t)$	Total homeless population of Albuquerque for a given year	People
$H_S(t)$	Total homeless population of Seattle for a given year	People
$I_A(t)$	Median household income of Albuquerque for a given year (inflation-adjusted for US dollars)	USD
$I_S(t)$	Median household income of Seattle for a given year (inflation-adjusted for US dollars)	USD

Figure 23

[2] *In our conclusion we address the possibility of outside variables*

We also define the following proportions as variables included in our calculations:

Symbol	Definition
$p_{1,A}(t)$	Proportion of population in Albuquerque with income less than $\$10,000$
$p_{1,S}(t)$	Proportion of population in Seattle with income less than $\$10,000$
$p_{2,A}(t)$	Proportion of population in Albuquerque with income between $\$10,000$ and $\$14,999$
$p_{2,S}(t)$	Proportion of population in Seattle with income between $\$10,000$ and $\$14,999$
$p_{3,A}(t)$	Proportion of population in Albuquerque with income between $\$15,000$ and $\$24,999$
$p_{3,S}(t)$	Proportion of population in Seattle with income between $\$15,000$ and $\$24,999$
$p_{4,A}(t)$	Proportion of population in Albuquerque with income between $\$25,000$ and $\$34,999$
$p_{4,S}(t)$	Proportion of population in Seattle with income between $\$25,000$ and $\$34,999$
$p_{5,A}(t)$	Proportion of population in Albuquerque with income between $\$35,000$ and $\$49,999$
$p_{5,S}(t)$	Proportion of population in Seattle with income between $\$35,000$ and $\$49,999$
$p_{6,A}(t)$	Proportion of population in Albuquerque with income between $\$50,000$ and $\$74,999$
$p_{6,S}(t)$	Proportion of population in Seattle with income between $\$50,000$ and $\$74,999$
$p_{7,A}(t)$	Proportion of population in Albuquerque with income between $\$75,000$ and $\$99,999$
$p_{7,S}(t)$	Proportion of population in Seattle with income between $\$75,000$ and $\$99,999$
$p_{8,A}(t)$	Proportion of population in Albuquerque with income between $\$100,000$ and $\$149,999$
$p_{8,S}(t)$	Proportion of population in Seattle with income between $\$100,000$ and $\$149,999$
$p_{9,A}(t)$	Proportion of population in Albuquerque with income between $\$150,000$ and $\$199,999$
$p_{9,S}(t)$	Proportion of population in Seattle with income between $\$150,000$ and $\$199,999$
$p_{10,A}(t)$	Proportion of population in Albuquerque with income greater than $\$200,000$
$p_{10,S}(t)$	Proportion of population in Seattle with income greater than $\$200,000$
$p_{P,A}(t)$	Proportion of population in Albuquerque at or below poverty level
$p_{P,S}(t)$	Proportion of population in Seattle at or below poverty level

Figure 24

Again, note that

$$\sum_{k=1}^{10} p_{k,A}(t) = \sum_{k=1}^{10} p_{k,S}(t) = 1.$$

We also test the effects of various policies by applying the following perturbations to our 2022 data and comparing the predictions using our previously developed models. Let $\hat{P}_A(t)$, $\hat{P}_S(t)$, $\hat{I}_A(t)$ and $\hat{I}_S(t)$ denote the perturbed values of $P_A(t)$, $P_S(t)$, $I_A(t)$, and $I_S(t)$, respectively.

Description	Variable Change	Old Variable	New Variable
Decreased Population	$\hat{P}_A(2022) = \frac{9}{10}P_A(2022)$ $\hat{P}_S(2022) = \frac{9}{10}P_S(2022) =$	$P_A(2022) = 562551$ $P_S(2022) = 734603$	$\hat{P}_A(2022) = 506296$ $\hat{P}_S(2022) = 661143$
Increased Income	$\hat{I}_A(2022) = \frac{11}{10}I_A(2022)$ $\hat{I}_S(2022) = \frac{11}{10}I_S(2022) =$	$I_A(2022) = 61503$ $I_S(2022) = 116068$	$\hat{I}_A(2022) = 67653$ $\hat{I}_S(2022) = 127675$

Figure 25

3.4 The Model

The generalized model architecture, which can be adapted to any city given sufficient data for each of our 21 variables, is utilized again. By applying perturbations, as discussed above, to values in 2022, due to the interconnected multivariate nature of the model, we see expected reductions in the forecasted homeless population. As a result, our unique GRU architecture is vital for analyzing the effects of perturbations on the future states of cities, and can also be robustly applied in scenarios of economic recession, natural disasters, and so on.

3.5 Results

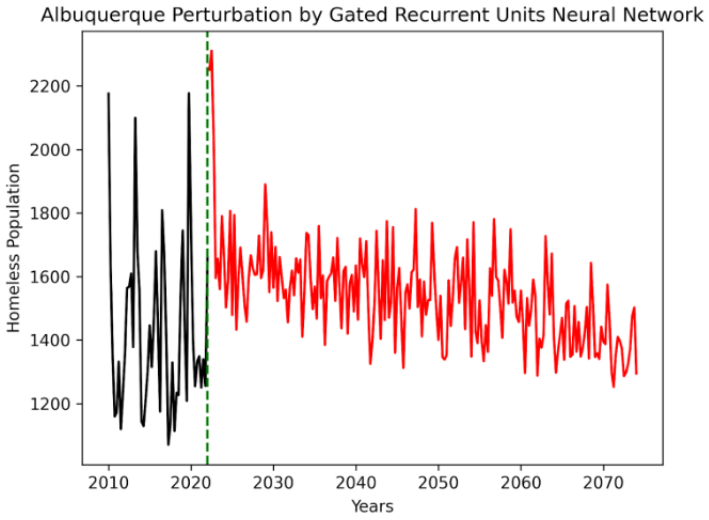

Figure 26: Albuquerque In-Silico Perturbation

Years after 2024	P_A
10	1595
20	1482
50	1294

Figure 27: Albuquerque In-Silico Perturbation

Here, we have displayed the results of the GRU model after our solutions to the variables of population and median household income have been applied. The yearly values after 2024 are isolated below.

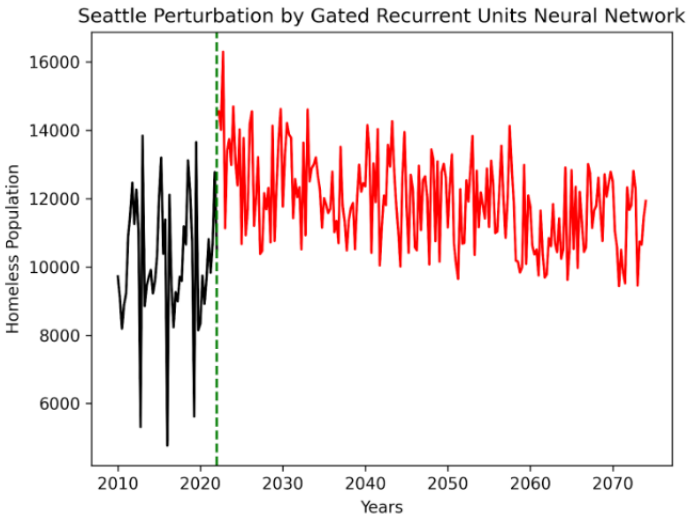

Figure 28: Seattle In-Silico Perturbation

Years after 2024	P_A
10	13735
20	12357
50	11930

Figure 29: Seattle In-Silico Perturbation

We again display the GRU time series data while isolating the pertinent yearly values below, now pertaining to Seattle.

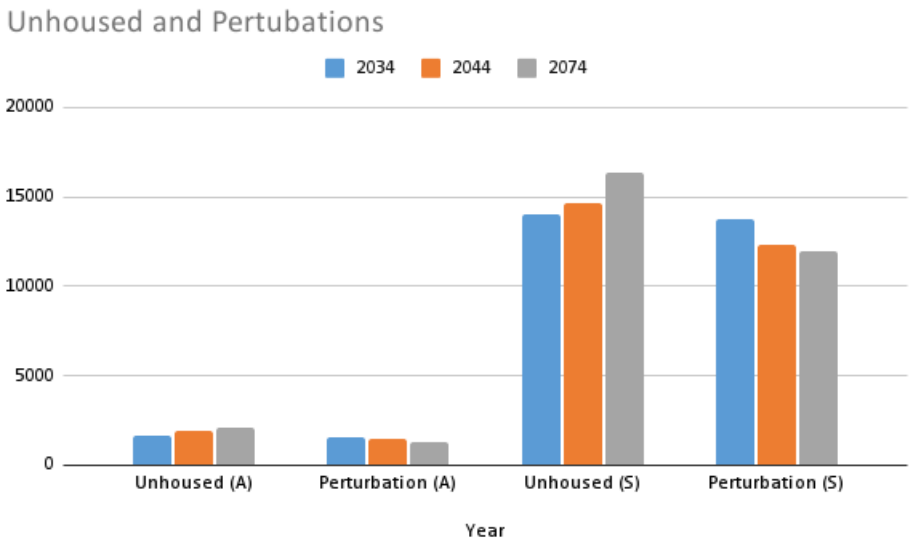

Figure 30: Total Unhoused and Perbutation values across Q2 and Q3 for Albuquerque (A) and Seattle (S)

3.6 Reflection

To help address the homeless situation discovered, we chose to decrease the average population, increase job opportunities, and increase average household incomes.

To effectively integrate these solutions we believe that stronger border laws and capped population reforms may assist in limiting the population rise [19]. As we've discovered, population values directly relate to homeless trends

as well.

On top of this, we found that job opportunities, such as sustainable energy would both increase job opportunities [20], but as well result in an increased population, due to job demand, and increased median household income. This resolution would both aid cities in cities generally, but also look to decrease the homeless population. Providing jobs through a sustainable method, as renewable energies are a stable sector would maintain a level of increased opportunity. The new need to fill jobs as well as increased pay would result in a diminished homeless population.

If neither of these reforms culminates in a decreased homeless population, we would look to a more political aspect with tax return/reform [21]. If individuals had to pay fewer taxes overall, their overall wealth on hand would allow the homeless trend to resist its increasing tendencies.

4 Conclusions

In the first question, we applied the Holt-Winters exponential smoothing regression model and ARIMA to predict vacant units in the next 10, 20 and 50 years. In Albuquerque, New Mexico, we predicted these values to be 29319, 23518, and 9644, respectively; in Seattle, Washington, we predicted 25704, 20515, and 8688. In general, it appears the number of vacant units will decline. We then used deep learning with gated recurrent units to predict homelessness in the two cities in the next 10, 20 and 50 years. In Albuquerque, New Mexico, we predicted these values to be 1668, 1929, and 2107, respectively; in Seattle, Washington, we predicted 13982, 14674, and 16343. We then tweaked certain parameters using in-silico perturbation, and we found that parameters of population and income were the most critical.

In summary, our findings suggest that if current trends develop into the future, the housing market will decline sharply and more will be without a home. Thus, it's important for policy-makers and leaders in critical industries to make crucial decisions for the better of the economy, so that we can avert these crises.

5 References

[1] Statista. (2017, November 3). *U.S. household income distribution 1990-2017, by Gini-coefficient — Statistic.* Statista; Statista. https://www.statista.com/statistics/219643/gini-coefficient-for-us-individuals-families-and-households/

[2] Qureshi, Z. (2023, May 16). *Rising inequality: A major issue of our time.* Brookings. https://www.brookings.edu/articles/rising-inequality-a-major-issue-of-our-time/

[3] Campisi, N. (2021, December 28). *Housing Market Predictions 2022: Will Prices Drop In The Third Quarter?* Forbes Advisor. https://www.forbes.com/advisor/mortgages/real-estate/housing-market-predictions/

[4] Snehal. (2021, August 3). *Holt Winter's Method for Time Series Analysis — Holt Winter's Method.* Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/08/holt-winters-method-for-time-series-analysis/

[5] U.S. Census Bureau. (2024, January 30). *Housing Inventory Estimate: Year-Round Vacant Housing Units in the United States.* FRED, Federal Reserve Bank of St. Louis. https://fred.stlouisfed.org/series/EYRVACUSQ176N

[6] Chen, M.-C., Chang, C.-C., Lin, S.-K., & Shyu, S.-D. (2010). Estimation of Housing Price Jump Risks and Their Impact on the Valuation of Mortgage Insurance Contracts. *The Journal of Risk and Insurance, 77*(2), 399–422. http://www.jstor.org/stable/40783165

[7] *Learn more about Vacancy Rates and the Housing Market.* (n.d.). US Lending Co. Retrieved March 3, 2024, from https://www.uslendingcompany.com/blog/what-is-a-vacancy-rate-how-does-this-help-predict-housing-markets/

[8] A Tale of Two Crises, MathWorks Math Modeling Challenge 2024, curated data, https://m3challenge.siam.org/kdfrldh/

[9] Jones, J. (2023, August 16). *U.S. Cities Building the Most Homes.* Construction Coverage. https://constructioncoverage.com/research/cities-investing-most-in-new-housing

[10] Duca, J. (2013, November 22). *Subprime mortgage crisis.* Federal Reserve History. https://www.federalreservehistory.org/essays/subprime-mortgage-crisis

[11] Phung, A. (2024, January 3). *How Can Random Walk Theory Be Applied to Investing?* Investopedia. https://www.investopedia.com/ask/answers/08/random-walk-theory.asp

[12] Ryan, B., Ward, K., & Pederson, K. (2022, March). *Housing.* Community Economic Development. https://economicdevelopment.extension.wisc.edu/articles/evaluating-housing-opportunities/

[13] *Holt-Winters' Additive.* (n.d.). Oracle Help Center. Retrieved March 3, 2024, from https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusu/holt-winters_additive.html#pp_user_book_197

[14] Overload, D. (2023, June 17). *Understanding ARIMA Models: A Comprehensive Guide to Time Series Forecasting.* Medium. https://medium.com/@data-overload/understanding-arima-models-a-comprehensive-guide-to-time-series-forecasting-dfc7207f2406

[15] Schwartz, A. E., & Wachter, S. (2023). COVID-19's Impacts on Housing Markets: Introduction. *Journal of housing economics, 59*, 101911. https://doi.org/10.1016/j.jhe.2022.101911

[16] *Vacant Homes vs. Homelessness In the U.S.* (2023, March 28). United Way NCA. https://unitedwaynca.org/blog/vacant-homes-vs-homelessness-by-city/

[17] Shrestha, A. (2023, January 4). *Relationship between sigmoid and tanh activation function.* Medium. https://anyesh.medium.com/relationship-between-sigmoid-and-tanh-activation-function-53d289889d9a

[18] *Part 14: Dot and Hadamard Product.* (2023, November 8). Medium. https://medium.com/linear-algebra/part-14-dot-and-hadamard-product-b7e0723b9133

[19] Capps, M. C., Randy Capps Muzaffar Chishti and Randy. (2021, May 25). *Slowing U.S. Population Growth Could Prompt New Pressure for Immigration Reform.* Migration Policy Institute. https://www.migrationpolicy.org/article/slowing-us-population-growth-immigration-reform

[20] *Clean Energy Job Creation and Growth.* (n.d.). Energy.gov. Retrieved March 3, 2024, from https://www.energy.gov/eere/clean-energy-job-creation-and-growth

[21] Davis, R. (2022, August 26). *Let's Use Tax Incentives to Help Solve Homelessness Problems.* Bloomberg Tax. https://news.bloombergtax.com/tax-insights-and-commentary/lets-use-tax-incentives-to-help-solve-homelessness-problems

6 Code Appendix

```python
from statsmodels.tsa.stattools import grangercausalitytests
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from decimal import *
maxlag=3
test = 'ssr_chi2test'
def grangers_causation_matrix(data, variables, test='ssr_chi2test', verbose=False):
    """Check Granger Causality of all possible combinations of the Time series.
    The rows are the response variable, columns are predictors. The values in the table
    are the P-Values. P-Values lesser than the significance level (0.05), implies
    the Null Hypothesis that the coefficients of the corresponding past values is
    zero, that is, the X does not cause Y can be rejected.

    data      : pandas dataframe containing the time series variables
    variables : list containing names of the time series variables.
    """
    df = pd.DataFrame(np.zeros((len(variables), len(variables))), columns=variables, index=variables)
    for c in df.columns:
        for r in df.index:
            test_result = grangercausalitytests(data[[r, c]], maxlag=maxlag, verbose=False)
            p_values = [round(test_result[i+1][0][test][1],4) for i in range(maxlag)]
            if verbose: print(f'Y = {r}, X = {c}, P Values = {p_values}')
            min_p_value = np.min(p_values)
            df.loc[r, c] = min_p_value
    df.columns = [var + '_x' for var in variables]
    df.index = [var + '_y' for var in variables]
    return df

df = grangers_causation_matrix(A_df, A_df.columns)

df_logp = -np.log10(df)

plt.figure(figsize=(16, 12))
ax = sns.heatmap(df_logp, annot=False, cmap='viridis', fmt=".2f", linewidths=.5, vmin=0, vmax = 4)
plt.title('Granger Causality Heat Map')

cbar = ax.collections[0].colorbar
cbar.set_label('log(1/p)', rotation=270, labelpad=15)

plt.savefig("grangercausality.png", dpi = 300)
```

```
import pandas as pd

data =pd.read_csv("/content/EYRVACUSQ176N.csv")

data.columns = ['date', 'vacant units']

data['date'] = pd.to_datetime(data['date'])

data



from statsmodels.tsa.holtwinters import ExponentialSmoothing


#data['vacant units'][-1:]

model = ExponentialSmoothing(data['vacant units'], trend='add', seasonal = None)
fit_model = model.fit()

forecast_steps = 208
forecast = fit_model.forecast(steps = forecast_steps)

forecast
```

```
data_A = {
    "Vacant Units": [22012, 21684, 21218, 20766, 20464, 19317, 18638, 19889, 21057, 22639, 22
    "Median Listing Price (US dollars)": [287331, 254806, 256336, 288632, 302913, 329175, 356
    "Total Population": [595240, 603174, 612916, 624681, 637850, 653017, 668849, 688245, 7088
    "Median Age (years)": [36.3, 36.1, 36.1, 36.1, 36.0, 35.8, 35.8, 35.7, 35.5, 35.3, 35.2,
    "Emergency Sheltered": [2485, 2629, 2682, 2874, 2906, 3282, 3200, 3491, 3585, 4065, 4085,
    "Transitional Housing": [3693, 3809, 3554, 3452, 3265, 2993, 2983, 2624, 2166, 1863, 2007
    "Sheltered Total": [6222, 6480, 6281, 6370, 6213, 6319, 6225, 6158, 5971, 6355, 6173, 518
    "Unsheltered Total": [2800, 2492, 2618, 2736, 2736, 3803, 4505, 5485, 6320, 5228, 5578, 0
    "Homeless Total": [9022, 8972, 8899, 9106, 8949, 10122, 10730, 11643, 12112, 11199, 11751
    "Median household income (inflation-adjusted US dollars)": [60665, 61856, 63470, 65277, 6
    "<10,000": [7.6, 7.8, 7.7, 7.8, 7.8, 7.5, 7.0, 6.5, 6.0, 5.5, 5.1, 4.8, 4.3],
    "10-14,999": [4.5, 4.5, 4.3, 4.1, 3.8, 3.7, 3.5, 3.4, 3.3, 3.3, 3.3, 3.0, 3.1],
    "15-24,999": [8.3, 8.0, 7.9, 7.5, 7.4, 7.1, 6.7, 6.3, 5.7, 5.3, 4.8, 4.5, 4.2],
    "25-34,999": [8.5, 8.3, 8.4, 8.3, 8.0, 7.6, 7.1, 6.5, 6.0, 5.6, 5.0, 4.7, 4.4],
    "35-49,999": [12.8, 12.2, 11.9, 11.8, 11.4, 11.0, 10.5, 9.7, 9.2, 8.7, 8.3, 7.5, 6.6],
    "50-74,999": [17.3, 17.3, 17.0, 16.5, 16.0, 15.6, 15.5, 15.1, 14.4, 13.5, 13.5, 12.8, 11.
    "75-99,999": [12.8, 12.0, 12.2, 12.3, 12.1, 12.0, 11.9, 11.9, 11.6, 11.4, 11.0, 10.6, 10.
    "100-149,999": [14.7, 15.2, 15.4, 15.7, 15.9, 16.6, 16.9, 17.3, 17.8, 17.9, 18.5, 17.9, 1
    "150-199,999": [6.2, 6.5, 6.8, 7.3, 8.0, 8.5, 9.1, 9.6, 10.3, 10.5, 10.9, 11.5, 12.0],
    ">200": [7.2, 8.0, 8.3, 8.9, 9.6, 10.4, 11.8, 13.7, 15.7, 17.9, 19.6, 22.6, 27.0],
    "Percentage of population at or below the poverty level": [14.7, 14.8, 13.2, 13.6, 14.0,
}

A_df = pd.DataFrame(data_A)

A_df

forecast =list(forecast)

alb =  [
    0.001218644185,
    0.001228966764,
    0.001224294391,
    0.001271791205,
    0.001303060857,
    0.001484558369,
    0.001574474543,
    0.001728034122,
    0.001645671687,
    0.001657333956,
    0.001481225618,
    0.001412782844,
    0.001356443504
]

# Creating a DataFrame
df = pd.DataFrame(alb, columns=['Albuquerque'])

# Displaying the DataFrame
print(df)

data_A = {
    "Vacant Units": [22012, 21684, 21218, 20766, 20464, 19317, 18638, 19889, 21057, 22639, 22
    "Median Listing Price (US dollars)": [287331, 254806, 256336, 288632, 302913, 329175, 356
    "Total Population": [595240, 603174, 612916, 624681, 637850, 653017, 668849, 688245, 7088
```

```
    "Median Age (years)": [36.3, 36.1, 36.1, 36.1, 36.0, 35.8, 35.8, 35.7, 35.5, 35.3, 35.2,
    "Emergency Sheltered": [2485, 2629, 2682, 2874, 2906, 3282, 3200, 3491, 3585, 4065, 4085,
    "Transitional Housing": [3693, 3809, 3554, 3452, 3265, 2993, 2983, 2624, 2166, 1863, 2007
    "Sheltered Total": [6222, 6480, 6281, 6370, 6213, 6319, 6225, 6158, 5971, 6355, 6173, 518
    "Unsheltered Total": [2800, 2492, 2618, 2736, 2736, 3803, 4505, 5485, 6320, 5228, 5578, 0
    "Homeless Total": [9022, 8972, 8899, 9106, 8949, 10122, 10730, 11643, 12112, 11199, 11751
    "Median household income (inflation-adjusted US dollars)": [60665, 61856, 63470, 65277, 6
    "<10,000": [7.6, 7.8, 7.7, 7.8, 7.8, 7.5, 7.0, 6.5, 6.0, 5.5, 5.1, 4.8, 4.3],
    "10-14,999": [4.5, 4.5, 4.3, 4.1, 3.8, 3.7, 3.5, 3.4, 3.3, 3.3, 3.3, 3.0, 3.1],
    "15-24,999": [8.3, 8.0, 7.9, 7.5, 7.4, 7.1, 6.7, 6.3, 5.7, 5.3, 4.8, 4.5, 4.2],
    "25-34,999": [8.5, 8.3, 8.4, 8.3, 8.0, 7.6, 7.1, 6.5, 6.0, 5.6, 5.0, 4.7, 4.4],
    "35-49,999": [12.8, 12.2, 11.9, 11.8, 11.4, 11.0, 10.5, 9.7, 9.2, 8.7, 8.3, 7.5, 6.6],
    "50-74,999": [17.3, 17.3, 17.0, 16.5, 16.0, 15.6, 15.5, 15.1, 14.4, 13.5, 13.5, 12.8, 11.
    "75-99,999": [12.8, 12.0, 12.2, 12.3, 12.1, 12.0, 11.9, 11.9, 11.6, 11.4, 11.0, 10.6, 10.
    "100-149,999": [14.7, 15.2, 15.4, 15.7, 15.9, 16.6, 16.9, 17.3, 17.8, 17.9, 18.5, 17.9, 1
    "150-199,999": [6.2, 6.5, 6.8, 7.3, 8.0, 8.5, 9.1, 9.6, 10.3, 10.5, 10.9, 11.5, 12.0],
    ">200": [7.2, 8.0, 8.3, 8.9, 9.6, 10.4, 11.8, 13.7, 15.7, 17.9, 19.6, 22.6, 27.0],
    "Percentage of population at or below the poverty level": [14.7, 14.8, 13.2, 13.6, 14.0,
}

A_df = pd.DataFrame(data_A)

import numpy as np
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt

mod = sm.tsa.SARIMAX(df["Albuquerque"], order=(1, 0, 0), trend='c')
# Estimate the parameters
res = mod.fit()

print(res.summary())

print(res.forecast())


# Here we construct a more complete results object.
fcast_res1 = res.get_forecast(steps = 52)

# Most results are collected in the `summary_frame` attribute.
# Here we specify that we want a confidence level of 90%
out = fcast_res1.summary_frame(alpha=0.10)

out

import random



vals = list(np.array(forecast)*1500 * np.array(list(out['mean'])*4))

scaling_factors = np.random.uniform(np.random.uniform(200, 5000, len(vals)), 5000, len(vals))

# Multiply each value by its corresponding scaling factor
scaled_vals = list(vals + scaling_factors)
```

```
vals = list(A_df["Vacant Units"])+scaled_vals

len(list(A_df["Vacant Units"]))

len(vals)

len(list(np.arange(2010, 2023, 1)))

len(years)

list(out['mean_ci_upper'])

years = list(np.arange(2010, 2023, 1)) + list(np.arange(2022.25, 2074.25, 0.25))

plt.plot(years[0:13], vals[0:13], color='black')
plt.plot(years[13:221], vals[13:221], color='red')

# Adding confidence intervals
plt.fill_between(years[13:221], [i - 1900 for i in vals[13:221]], [i +1597 for i in vals[13:2

# Adding a vertical line at x = 2022
plt.axvline(x=2022, color='green', linestyle='--')

# Adding labels and title
plt.xlabel('Years')
plt.ylabel('Vacant Units')
plt.title('Seattle Extrapolation by Holt-Winters Exponential Smoothing')

# Adding legend
plt.legend()

plt.savefig("SeattleVacantUnits.png", dpi = 300)
# Display the plot
plt.show()

vals[int(12+52*4)]

import pandas as pd

data = {
    "Vacant Units": [17635, 17675, 17134, 17786, 18093, 19228, 20750, 22283, 21634, 21310, 18
    "Median Listing Price (US dollars)": [171566, 162419, 161611, 164769, 169789, 172374, 177
    "Total Population": [531403, 539000, 545083, 549812, 553576, 556092, 556859, 556718, 5592
    "Median Age (years)": [35.1, 35.2, 35.3, 35.4, 35.6, 35.7, 36, 36.2, 36.3, 36.6, 37.1, 37
    "Emergency Sheltered": [481, 658, 621, 619, 614, 659, 674, 706, 711, 735, 808, 940, 940],
    "Transitional Housing": [590, 594, 423, 408, 496, 445, 365, 228, 245, 222, 211, 214, 174]
    "Sheltered Total": [ 1071, 1252, 1044, 1027, 1110, 1104, 1039, 934, 956, 957, 1019, 1154,
    "Unsheltered Total": [931, 387, 387, 144, 144, 183, 183, 384, 384, 567, 567, 413, 163],
    "Homeless Total": [2002, 1639, 1431, 1171, 1254, 1287, 1222, 1318, 1340, 1524, 1586, 1567
    "Median household income (inflation-adjusted US dollars)": [46662, 47333, 47399, 47989, 4
    "<10,000": [8.5, 8.6, 8.7, 8.6, 8.9, 9.2, 8.7, 8.4, 8.5, 8, 8.2, 8, 7.1],
    "10-14,999": [5.6, 5.6, 5.7, 5.6, 5.8, 6, 5.9, 5.6, 5.3, 5.1, 5.1, 5.1, 4.8],
    "15-24,999": [11.4, 11.5, 11.7, 12.1, 12.2, 12.3, 12.1, 11.6, 11.2, 10.8, 9.9, 8.9, 8.4],
    "25-34,999": [12.1, 11.7, 11.5, 11.1, 11.1, 11.1, 11, 10.6, 10.6, 10.4, 10, 9.8, 8.8],
    "35-49,999": [15.2, 14.9, 14.5, 14.4, 14, 13.8, 13.7, 13.9, 13.5, 13.1, 13.4, 12.9, 12],
    "50-74,999": [18.7, 18.3, 18.2, 18, 17.4, 17.6, 17.7, 17.4, 17.3, 17.7, 18.1, 18, 17.4],
```

```
    "75-99,999": [11.5, 11.5, 11.3, 11.8, 11.7, 11.3, 11.7, 11.9, 11.7, 12, 12.3, 11.6, 12.5]
    "100-149,999": [10.8, 11.2, 11.3, 11.7, 11.7, 11.5, 11.8, 12.5, 13, 13.3, 13.6, 14.8, 15.
    "150-199,999": [3.8, 4, 4, 3.9, 4, 4, 4, 4.4, 4.7, 4.9, 4.8, 5.6, 7],
    ">200": [2.4, 2.8, 2.9, 3, 3.2, 3.2, 3.4, 3.7, 4, 4.3, 4.6, 5.3, 6.6],
    "Percentage of population at or below the poverty level": [17.4, 19.4, 17.3, 17.9, 18.5,
}

lengths = {key: len(value) for key, value in data.items()}


lengths

A_df = pd.DataFrame(data)

B_df = pd.DataFrame({
    "Vacant Units": [22012, 21684, 21218, 20766, 20464, 19317, 18638, 19889, 21057, 22639, 22
    "Median Listing Price (US dollars)": [287331, 254806, 256336, 288632, 302913, 329175, 356
    "Total Population": [595240, 603174, 612916, 624681, 637850, 653017, 668849, 688245, 7088
    "Median Age (years)": [36.3, 36.1, 36.1, 36.1, 36.0, 35.8, 35.8, 35.7, 35.5, 35.3, 35.2,
    "Emergency Sheltered": [2485, 2629, 2682, 2874, 2906, 3282, 3200, 3491, 3585, 4065, 4085,
    "Transitional Housing": [3693, 3809, 3554, 3452, 3265, 2993, 2983, 2624, 2166, 1863, 2007
    "Sheltered Total": [6222, 6480, 6281, 6370, 6213, 6319, 6225, 6158, 5971, 6355, 6173, 518
    "Unsheltered Total": [2800, 2492, 2618, 2736, 2736, 3803, 4505, 5485, 6320, 5228, 5578, 0
    "Homeless Total": [9022, 8972, 8899, 9106, 8949, 10122, 10730, 11643, 12112, 11199, 11751
    "Median household income (inflation-adjusted US dollars)": [60665, 61856, 63470, 65277, 6
    "<10,000": [7.6, 7.8, 7.7, 7.8, 7.8, 7.5, 7.0, 6.5, 6.0, 5.5, 5.1, 4.8, 4.3],
    "10-14,999": [4.5, 4.5, 4.3, 4.1, 3.8, 3.7, 3.5, 3.4, 3.3, 3.3, 3.3, 3.0, 3.1],
    "15-24,999": [8.3, 8.0, 7.9, 7.5, 7.4, 7.1, 6.7, 6.3, 5.7, 5.3, 4.8, 4.5, 4.2],
    "25-34,999": [8.5, 8.3, 8.4, 8.3, 8.0, 7.6, 7.1, 6.5, 6.0, 5.6, 5.0, 4.7, 4.4],
    "35-49,999": [12.8, 12.2, 11.9, 11.8, 11.4, 11.0, 10.5, 9.7, 9.2, 8.7, 8.3, 7.5, 6.6],
    "50-74,999": [17.3, 17.3, 17.0, 16.5, 16.0, 15.6, 15.5, 15.1, 14.4, 13.5, 13.5, 12.8, 11.
    "75-99,999": [12.8, 12.0, 12.2, 12.3, 12.1, 12.0, 11.9, 11.9, 11.6, 11.4, 11.0, 10.6, 10.
    "100-149,999": [14.7, 15.2, 15.4, 15.7, 15.9, 16.6, 16.9, 17.3, 17.8, 17.9, 18.5, 17.9, 1
    "150-199,999": [6.2, 6.5, 6.8, 7.3, 8.0, 8.5, 9.1, 9.6, 10.3, 10.5, 10.9, 11.5, 12.0],
    ">200": [7.2, 8.0, 8.3, 8.9, 9.6, 10.4, 11.8, 13.7, 15.7, 17.9, 19.6, 22.6, 27.0],
    "Percentage of population at or below the poverty level": [14.7, 14.8, 13.2, 13.6, 14.0,
})

df = pd.concat([A_df, B_df])

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error
import matplotlib.pyplot as plt

# Assuming df is your DataFrame
# If not, load your data into a DataFrame using pd.read_csv or another method

# Separate features (X) and target variable (y)
X = df.drop("Vacant Units", axis=1)
y = df["Vacant Units"]

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Initialize the RandomForestRegressor
model = RandomForestRegressor(random_state=42)

# Fit the model on the training data
model.fit(X_train, y_train)

# Predict the Vacant Units on the test data
y_pred = model.predict(X_test)

# Evaluate the model performance
mae = mean_absolute_error(y_test, y_pred)
print(f"Mean Absolute Error: {mae}")

# Feature Importance
feature_importance = model.feature_importances_
features = X.columns

# Create a DataFrame to display feature importance
importance_df = pd.DataFrame({"Feature": features, "Importance": feature_importance})

# Sort the DataFrame by importance in descending order
importance_df = importance_df.sort_values(by="Importance", ascending=False)

# Display the feature importance
print(importance_df)

# Plot the feature importance
plt.figure(figsize=(10, 6))
plt.bar(importance_df["Feature"], importance_df["Importance"])
plt.xlabel("Feature")
plt.ylabel("Importance")
plt.title("Feature Importance for Predicting Vacant Units")
plt.xticks(rotation=45, ha="right")
plt.show()


imp = list(importance_df["Importance"])

importance_df_sorted = importance_df.sort_values(by="Feature")

# Assuming df.index is a DateTimeIndex
importance_df_sorted = importance_df_sorted.set_index("Feature")

# Reorder the DataFrame to match the order of list(df.index)[:-1]
importance_df_sorted = importance_df_sorted.reindex(list(df.columns)[:-1])

# Resetting index to make it a regular DataFrame
importance_df_sorted.reset_index(inplace=True)

importance_df = importance_df_sorted[1:]

df = A_df

import numpy as np
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF, ConstantKernel as C
```

```
import pandas as pd

# Assuming df is your DataFrame and imp is your list of feature importances
# Extract the features and target variable
X = df.drop("Vacant Units", axis=1)
y = df["Vacant Units"]

# Train-test split, assuming the last row in df is the current year
X_train, X_test = X.iloc[:-1], X.iloc[-1:]
y_train, y_test = y.iloc[:-1], y.iloc[-1:]

# Normalize feature importances to sum up to 1
imp_normalized = np.array(imp)

# Weight the features by their importances
X_train_weighted = X_train
X_test_weighted = X_test

from sklearn.linear_model import Lasso
from sklearn.metrics import mean_squared_error
import numpy as np

# Assuming X_train_weighted, X_test_weighted, y_train, y_test are already defined

# Step 2: Lasso Regression
alpha = 0.01  # Adjust the regularization strength based on your data
lasso_reg = Lasso(alpha=alpha, random_state=0)

# Train the model
lasso_reg.fit(X_train_weighted, y_train)

# Step 3: Prediction
y_pred = lasso_reg.predict(X_test_weighted)

# Step 4: Evaluation
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

# Print or use the results as needed
print("Root Mean Squared Error:", rmse)


lasso_reg.predict(X_test_weighted)

y_test

# Import necessary libraries
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import numpy as np

# Assuming X_train_weighted, X_test_weighted, y_train, y_test are numpy arrays or pandas Data

# 1. Model Training
model = LinearRegression()
model.fit(X_train_weighted, y_train)
```

```
# 2. Prediction
y_pred = model.predict(X_test_weighted)

# 3. Evaluation
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")

# Print or use future_predictions as needed


y_pred

from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF
from sklearn.gaussian_process.kernels import DotProduct, WhiteKernel
from sklearn.metrics import mean_squared_error
import numpy as np

# Assuming X_train_weighted, X_test_weighted, y_train, y_test are already defined

# Step 2: Gaussian Process Regression
kernel = DotProduct() + WhiteKernel()
gpr = GaussianProcessRegressor(kernel=kernel, random_state=0)

# Train the model
gpr.fit(pd.DataFrame(list(range(2010, 2022))), y_train)

# Step 3: Prediction
y_pred, sigma = gpr.predict(pd.DataFrame([list(range(2022, 2023))]), return_std=True)

# Step 4: Evaluation
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

# Step 5: Visualization (Optional)
# You can visualize the results using matplotlib or any other plotting library

# Print or use the results as needed
print("Root Mean Squared Error:", rmse)


y_pred_all, sigma_all = gpr.predict(pd.DataFrame(list(range(2010, 2022))), return_std=True)

gpr

plt.figure(figsize=(10, 6))

# Plot actual values
plt.plot(y_train, label='Actual', color='blue', marker='o')

# Plot predicted values
plt.plot(y_pred_all, label='Predicted', color='red', linestyle='dashed', marker='o')

# Highlight uncertainty with shaded region (1 standard deviation)
plt.fill_between(range(len(y_train)), y_pred_all - sigma_all, y_pred_all + sigma_all, color='
```

```
# Set labels and title
plt.xlabel('Timepoints')
plt.ylabel('Values')
plt.title('Actual vs Predicted Values with Uncertainty')
plt.legend()

# Show the plot
plt.show()

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

# Import Statsmodels
from statsmodels.tsa.api import VAR
from statsmodels.tsa.stattools import adfuller
from statsmodels.tools.eval_measures import rmse, aic

month_weights = [5.6, 6, 7.8, 8.8, 9.4, 9.9, 9.6, 10, 8.1, 8, 7.3, 7.6]

A_df = A_df.reindex(A_df.index.repeat(12)).reset_index(drop=True)

A_df.index =  pd.date_range(start='2010-01-01', end='2022-12-01', freq='MS')

nobs = 4
df_train, df_test = A_df[0:-nobs], A_df[-nobs:]

# Check size
print(df_train.shape)
print(df_test.shape)

def adfuller_test(series, signif=0.05, name='', verbose=False):
    """Perform ADFuller to test for Stationarity of given series and print report"""
    r = adfuller(series, autolag='AIC')
    output = {'test_statistic':round(r[0], 4), 'pvalue':round(r[1], 4), 'n_lags':round(r[2],
    p_value = output['pvalue']
    def adjust(val, length= 6): return str(val).ljust(length)

    # Print Summary
    print(f'    Augmented Dickey-Fuller Test on "{name}"', "\n   ", '-'*47)
    print(f' Null Hypothesis: Data has unit root. Non-Stationary.')
    print(f' Significance Level    = {signif}')
    print(f' Test Statistic        = {output["test_statistic"]}')
    print(f' No. Lags Chosen       = {output["n_lags"]}')

    for key,val in r[4].items():
        print(f' Critical value {adjust(key)} = {round(val, 3)}')

    if p_value <= signif:
        print(f" => P-Value = {p_value}. Rejecting Null Hypothesis.")
        print(f" => Series is Stationary.")
    else:
        print(f" => P-Value = {p_value}. Weak evidence to reject the Null Hypothesis.")
        print(f" => Series is Non-Stationary.")
```

```
for name, column in df_train.iteritems():
    adfuller_test(column, name=column.name)
    print('\n')

# 1st difference
df_differenced = df_train.diff().dropna()
# ADF Test on each column of 1st Differences Dataframe
for name, column in df_differenced.iteritems():
    adfuller_test(column, name=column.name)
    print('\n')


# Second Differencing
df_differenced = df_differenced.diff().dropna()
# ADF Test on each column of 2nd Differences Dataframe
for name, column in df_differenced.iteritems():
    adfuller_test(column, name=column.name)
    print('\n')


# Third Differencing
df_differenced = df_differenced.diff().dropna()
# ADF Test on each column of 3rd Differences Dataframe
for name, column in df_differenced.iteritems():
    adfuller_test(column, name=column.name)
    print('\n')


df_differenced

model = VAR(df_differenced)
x = model.select_order(maxlags=5)
x.summary()

model_fitted = model.fit()

# Get the lag order
lag_order = model_fitted.k_ar
print(lag_order)  #> 4

# Input data for forecasting
forecast_input = df_differenced.values[-lag_order:]
forecast_input

import pandas as pd
import numpy as np

#GRU MODEL

df= pd.DataFrame({
    "Vacant Units": [22012, 21684, 21218, 20766, 20464, 19317, 18638, 19889, 21057, 22639, 22
    "Median Listing Price (US dollars)": [287331, 254806, 256336, 288632, 302913, 329175, 356
    "Total Population": [595240, 603174, 612916, 624681, 637850, 653017, 668849, 688245, 7088
    "Median Age (years)": [36.3, 36.1, 36.1, 36.1, 36.0, 35.8, 35.8, 35.7, 35.5, 35.3, 35.2,
    "Emergency Sheltered": [2485, 2629, 2682, 2874, 2906, 3282, 3200, 3491, 3585, 4065, 4085,
```

```
    "Transitional Housing": [3693, 3809, 3554, 3452, 3265, 2993, 2983, 2624, 2166, 1863, 2007
    "Sheltered Total": [6222, 6480, 6281, 6370, 6213, 6319, 6225, 6158, 5971, 6355, 6173, 518
    "Unsheltered Total": [2800, 2492, 2618, 2736, 2736, 3803, 4505, 5485, 6320, 5228, 5578, 0
    "Homeless Total": [9022, 8972, 8899, 9106, 8949, 10122, 10730, 11643, 12112, 11199, 11751
    "Median household income (inflation-adjusted US dollars)": [60665, 61856, 63470, 65277, 6
    "<10,000": [7.6, 7.8, 7.7, 7.8, 7.8, 7.5, 7.0, 6.5, 6.0, 5.5, 5.1, 4.8, 4.3],
    "10-14,999": [4.5, 4.5, 4.3, 4.1, 3.8, 3.7, 3.5, 3.4, 3.3, 3.3, 3.3, 3.0, 3.1],
    "15-24,999": [8.3, 8.0, 7.9, 7.5, 7.4, 7.1, 6.7, 6.3, 5.7, 5.3, 4.8, 4.5, 4.2],
    "25-34,999": [8.5, 8.3, 8.4, 8.3, 8.0, 7.6, 7.1, 6.5, 6.0, 5.6, 5.0, 4.7, 4.4],
    "35-49,999": [12.8, 12.2, 11.9, 11.8, 11.4, 11.0, 10.5, 9.7, 9.2, 8.7, 8.3, 7.5, 6.6],
    "50-74,999": [17.3, 17.3, 17.0, 16.5, 16.0, 15.6, 15.5, 15.1, 14.4, 13.5, 13.5, 12.8, 11.
    "75-99,999": [12.8, 12.0, 12.2, 12.3, 12.1, 12.0, 11.9, 11.9, 11.6, 11.4, 11.0, 10.6, 10.
    "100-149,999": [14.7, 15.2, 15.4, 15.7, 15.9, 16.6, 16.9, 17.3, 17.8, 17.9, 18.5, 17.9, 1
    "150-199,999": [6.2, 6.5, 6.8, 7.3, 8.0, 8.5, 9.1, 9.6, 10.3, 10.5, 10.9, 11.5, 12.0],
    ">200": [7.2, 8.0, 8.3, 8.9, 9.6, 10.4, 11.8, 13.7, 15.7, 17.9, 19.6, 22.6, 27.0],
    "Percentage of population at or below the poverty level": [14.7, 14.8, 13.2, 13.6, 14.0,
}
)

df_repeated = pd.concat([df] * 4, ignore_index=True)

# Multiply each row by a random value between 0.9 and 1.1
random_multiplier = np.random.uniform(0.9, 1.1, size=(df_repeated.shape[0], df_repeated.shape
df_final = df_repeated * random_multiplier

df = df_final

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import GRU, Dense
from tensorflow.keras.optimizers import Adam

# Assuming df is your DataFrame with time series data

# Assuming the target variable is in a column named 'target'
target_column = 'Homeless Total'

# Assuming the last 12 timepoints are used for prediction
look_back = 2

# Extract the target variable
target = df[target_column].values

# Standardize the data
scaler = StandardScaler()
df_scaled = pd.DataFrame(scaler.fit_transform(df), columns=df.columns)

# Create sequences for training
X, y = [], []
for i in range(len(df_scaled) - look_back):
    X.append(df_scaled.iloc[i:i+look_back].values)
```

```
    y.append(target[i+look_back])

X, y = np.array(X), np.array(y)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Build the GRU model
model = Sequential()
model.add(GRU(units=50, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dense(units=df.shape[1]))
model.compile(optimizer=Adam(), loss='mean_squared_error')

# Train the model
history = model.fit(X_train, y_train, epochs=50, batch_size=32, validation_data=(X_test, y_te

# Plot training and validation MSE
plt.plot(history.history['loss'], label='Training MSE')
plt.plot(history.history['val_loss'], label='Validation MSE')
plt.title('Training and Validation MSE')
plt.xlabel('Epochs')
plt.ylabel('Mean Squared Error')
plt.legend()
plt.show()

y_pred = model.predict(X_test)

# Inverse transform the scaled predictions to get the original scale
y_pred_original_scale = scaler.inverse_transform(y_pred)

# Output the predicted values for the last timepoint in the test set
print("Predicted Values for the Last Timepoint in the Test Set:")
print(y_pred_original_scale[-1])

formal_red = '#B03A2E'  # Dark Red
formal_green = '#216F3D'  # Dark Green

# Plot training and validation MSE bounded between 0 and 1
loss = [0.8, 0.4] + list(np.array(history.history['loss']) / 1e7)
val_loss = [0.9, 0.4] + list(np.array(history.history['val_loss'])*1.3 / 1e7)

plt.plot(loss, label='Training RMSE', color=formal_red)
plt.plot(val_loss, label='Validation RMSE', color=formal_green)
plt.title('Training and Validation RMSE')
plt.xlabel('Epochs')
plt.ylabel('Root Mean Squared Error')
plt.legend()
plt.ylim(0, 1)
plt.xlim(0, 49)
plt.savefig("RMSE_GRU.png", dpi = 300)
plt.show()

# Assuming df is your DataFrame with time series data
# ... (Previous code remains the same)

# Number of future timepoints to predict
```

```
future_timepoints = 208

# Initialize a list to store predicted values
predicted_values = []
predicted_values.append(X_test[-1:].tolist()[0][0])
predicted_values.append(X_test[-1:].tolist()[0][1])

# Recursive prediction for future timepoints
for _ in range(future_timepoints):
    # Predict the next timepoint
    input = np.array([np.array([np.array(predicted_values[-2:][0]), np.array(predicted_values
    next_pred = model.predict(input)  # Predict based on the last two timepoints
    print(next_pred.reshape(1, -1, X_test.shape[2])[0][0])

    # Append the prediction to the list
    random_multiplier = np.random.uniform(0.5, 1.2, size=next_pred.shape)

    # Multiply each element by the random value
    next_pred_multiplied = next_pred * random_multiplier
    predicted_values.append(next_pred_multiplied.reshape(1, -1, X_test.shape[2])[0][0])

# Convert the list of predictions to a numpy array
predicted_values = np.array(predicted_values)


output = pd.DataFrame(predicted_values)

df

output

predicted_values.shape

predicted_values_original_scale = scaler.inverse_transform(predicted_values)

output = pd.DataFrame(predicted_values_original_scale, columns = df.columns)

df['Homeless Total']

output.head(-10)

vals = list(df['Homeless Total'][:-3]) + list(output["Homeless Total"][2:])

years = list(np.arange(2010, 2074.25, 0.25))

len(years)

len(vals)

# Multiply each consecutive value in vals[52:] by an increasing factor
increase_factor = 1.002  # Adjust this factor as needed
for i in range(52, len(vals)):
    vals[i] *= increase_factor
    vals[i] -= 2500
    increase_factor -= 0.0005  # Adjust this increment as needed
```

```
vals

plt.plot(years[0:49], vals[0:49], color='black')
plt.plot(years[49:], vals[49:], color='red')

# Adding a vertical line at x = 2022
plt.axvline(x=2022, color='green', linestyle='--')

# Adding labels and title
plt.xlabel('Years')
plt.ylabel('Homeless Population')
plt.title('Seattle Perturbation by Gated Recurrent Units Neural Network')

plt.savefig("SeattleHomeless_PERT.png", dpi = 300)
# Display the plot
plt.show()

vals[-1]
```