MathWorks Math Modeling Challenge 2025

St. John's School

Team # 17524, Houston, Texas

Coach: Dwight Raulston

Students: Anik Banerji, David Qian, John Vu, Brandon Wu, Helen Yang

M3 Challenge THIRD PLACE—$10,000 Team Award

JUDGE COMMENTS

Specifically for Team #17524—Submitted at the close of triage judging

COMMENT 1: Nice sensitivity analysis for first model.

COMMENT 2: You submitted a well-written paper. You had original ideas in modeling all three parts of the challenge. You provided a concise and adequate summary. You did a great job listing your assumptions and their justification. More references could have been included in your justifications. You demonstrated a good understanding of Newton's law of heating to model the changes in the indoor temperature of a house. House types could be integrated into the model. Good job performing the sensitivity analysis as well as listing the strengths and weaknesses of your model.

In the second part, you did a good job applying the multivariate linear regression model. You included Mean Absolute Percent Error to measure the accuracy, finding out that the error was low. You identified the variability of peak temperature and peak load as a challenge to be addressed by finding more data and training the model over a longer period of time. You did a good job in scaling and normalizing all factors to model the vulnerability scores for selected Memphis ZIP codes. You generated heat maps and provided ranking of zip codes according to their vulnerability. You provided clear recommendations for the policy makers. Good job overall.

COMMENT 3: The paper successfully responded to the three challenge problems. The models were thoughtfully constructed and modeling choices explained well. The discussion on results, strengths and weaknesses, and sensitivity of the models were strong features of the paper. The paper could benefit from including an explicit example computing the vulnerability score for a particular zip code. Overall, I enjoyed reading your paper.

COMMENT 4: The executive summary is well-written and provides a detailed overview of the methods used and the results obtained. Summary is exceptional; it is concise, yet it entices the reader. Good job with assumption statements and their justifications. The team demonstrated a thorough understanding of the methods used. Very creative—your ideas are quite unique.

COMMENT 5: The first differential equation model is comprehensive and incorporating conduction, radiation, ventilation, and internal heat sources. The assumption of uniform indoor temperature, ignoring spatial gradients (e.g., multi-story layouts) could be improved. The second model of multivariate regression is robust and with backward selection and multicollinearity analysis. The linear assumptions may oversimplify non-linear trends (e.g., GDP vs. temperature interaction). The third weighted-sum model is practical with normalized factors (e.g., income, transportation) for actionable policy recommendations. The equal weighting of factors without empirical validation (e.g., assuming "elderly impact = income impact") could be improved upon for more insights

Heatwave Havoc: Modeling the Future of Urban Cooling and Power In Memphis-Executive Summary

In recent years, heat waves have emerged as one of the most devastating and increasingly frequent natural disasters, pushing communities, economies, and ecosystems to their limits. As we navigate a warming world, addressing the causes, impacts, and mitigation strategies of heat waves in cities such as Memphis becomes not just a scientific necessity but a moral and policy imperative.

The first part of this report models the temperature changes of various non-air-conditioned dwellings over the course of a day-long Memphis heatwave using a differential equation model. This model involves expressing the rate of change of the temperature as a sum of conduction, ventilation, solar radiation, and internal heating components. We found that the maximum internal temperature of a realistic dwelling during a heat wave can reach a dangerously high range of 97 to 99 degrees Fahrenheit. These results are fairly consistent throughout the non-air-conditioned four dwellings, regardless of their shade, floor size, and internal heat output. As such, the danger of high home-heats seems to persist regardless of the home's geographical or architectural circumstance, and lack of air conditioning remains a threatening issue.

The second part of the report forecasted peak power demand in Memphis over the next twenty years using a multivariate linear regression, which factored in historical data on climate patterns and economic conditions. After starting with seven potential predictors, we used a system of backward selection and multicollinearity analysis with correlation matrices, narrowing down our desired variables to just peak temperature in a year and GDP of Memphis. We then ran a multivariate regression, forecasting that peak electricity demand in Memphis for 2045 would be 3,357.72 mW. Since the increase in electricity demand is very small (only 120.58 mW over 20 forecasted years), we concluded that there would not be any major changes in the maximum demand for power through these 20 years apart from a very small potential increase.

The third and final section used information about 27 ZIP codes in the Memphis area to quantify factors involved in heat-risk with a vulnerability score. Four factors, each of which contribute to a community's vulnerability and do not affect each other, were chosen - the economy, population, age demographics, and transportation mode share of a neighborhood. These factors were normalized so that each factor ranged from 0 (very low risk) to 4 (very high risk) and combined using a weighted sum. ZIP codes with higher vulnerability scores, such as 38109, 38105, and 38111 were determined to be more heavily affected by heat waves; they require more assistance in preparing for and responding to the effects of heat waves. Based on the vulnerability scores calculated, the team recommends that neighborhoods such as South Memphis, the Medical Center, and Downtown be given priority assistance in mitigating the effects of extreme heat. Such areas are home to high populations of low-income Memphians, and they stand to suffer more when critical pieces of infrastructure, such as the municipal power grid, are shut down. The urgency of heat waves, especially in highly vulnerable areas, calls for a comprehensive effort to cool down the urban environment.

Table of Contents

Q1: Hot to Go

1.1 Defining the Problem

The first problem asks us to develop a model that tracks the indoor temperature of a Memphis house without air conditioning during a summer heat wave. Our model considers the outside temperature during an example Memphis heat wave, and compares dwellings without air conditioning in varying circumstances.

1.2 Assumptions

1.2.1. **Convection through walls is proportional to the base area of the unit.**

- Justification: Heat loss and gain through walls are largely dictated by surface area. Since taller buildings with the same base area have less exposed wall surface per unit volume, we assume that the base area serves as a reasonable approximation of heat transfer magnitude for most dwellings.

1.2.2. **Solar heat gain is determined by radiation penetrating through windows.**

- Justification: Direct and diffuse solar radiation contribute significantly to indoor temperature fluctuations. Since glass allows heat to enter while trapping infrared radiation, we include window surface area and solar heat in our model.

1.2.3. **Ventilation occurs through cracks, windows, and openings.**

- Justification: Air exchange through natural ventilation systems like vents and cracks affects temperature by transferring heat between indoor and outdoor environments.

1.2.4. **Heat sources inside the dwelling are limited to electronics and human occupants.**

- Justification: Internal heat generation stems primarily from electrical appliances and human metabolic activity. Other sources, such as combustion heating, are not explicitly modeled, as they are either negligible in well-insulated homes or user-controlled.

1.2.5. **Older houses have less insulation than newly constructed homes.**

- Justification: Advancements in building codes and materials have led to improved insulation standards over time. Older homes generally have higher thermal conductivity due to degraded or outdated insulation, leading to more heat exchange with the environment.

1.2.6. **The house is modeled as an object with a constant internal temperature.**

- Justification: Studies on the optimization of temperature uniformity have shown that natural convection can effectively maintain uniform temperatures in enclosed spaces, and that differences in temperature in different areas quickly become negligible.

1.2.7. **There are no other significant heat sources, and heat storage in walls and furniture is negligible.**

- Justification: While walls, floors, and furniture can absorb and release heat over time, we assume their thermal mass effects are small compared to other heat exchange processes. Their influence on short-term temperature variations is considered negligible.

1.2.8. **Solar radiation is only present during daylight hours (6 AM to 8 PM). Outside this range (6 PM to 6 AM), solar radiation is assumed to be zero.**

- Justification: This reflects the fact that the sun is not visible at night, so no direct solar radiation reaches the Earth's surface during these hours.

1.2-9. **Sinusoidal Variation During Daytime**

- Justification: Solar radiation follows a sinusoidal pattern during the day, peaking at noon and tapering off symmetrically in the morning and afternoon. The term (hour - 6) shifts the sine wave so that it starts at 6 AM. The divisor 14 scales the sine wave to fit within the 14-hour daylight period (6 AM to 8 PM).

1.3.1 Development of the Model

We used a differential equation to model the change in internal temperature of the house because heat transfer in this scenario closely follows Newton's Law of Heating (shown below), which states that the rate of temperature change is proportional to the difference between an object's temperature and its surroundings. Since the house interacts with external factors, a differential equation describes how these influences combine to affect indoor temperature over time. By assuming the house has a uniform internal temperature, we can represent its behavior as a first-order differential equation.

$$\frac{dT}{dt} = -k_{con}(T_i - T_{env}) + T_{in}$$

Newton's Law of Heating

Additionally, we considered using other models, specifically, autoregressive models. However, we realized AR models were not suitable since we lacked data on past temperatures and cannot predict future values due to this constraint. Furthermore, an autoregressive approach would not account for other factors like radiation, ventilation, or internal heat sources.

1.3.2 Variables

Symbol	**Value**	**Units**
T_i, T_{env}	Temperature inside house, temperature outside house	Fahrenheit
C	Thermal capacitance of house	J/K
k_{con}	Thermal conductivity constant	W/K
a_{solar}, I_{solar}	Solar heat gain coefficient, solar radiance value	Unitless
k_{vent}	Ventilation coefficient that accounts for air leakage	W/K
n_{ppl}	Number of people	People
P_{app}	Power output of all appliances	W

Figure 1

1.3.3 Explanation of Model

We model the rate of change of indoor temperature as a function of four key heat transfer mechanisms: conduction, radiation, ventilation, and internal heat sources. We first consider each component separately:

- According to Newton's Law of Heating, the rate of conductive heat transfer is proportional to the temperature difference, leading to the expression $-k_{con}(T_i - T_{env})$ for conductivity.

- Radiation is influenced by the solar radiation intensity and the absorption properties of the house. This is defined by the expression $a_{solae} * I_{solar}$.
- Heat loss or gain through ventilation depends on the temperature difference between the inside and outside, and this is represented in the equation, again by Newton's Law of Heating, $k_{vent}(T_i - T_{env})$. The sign is positive because if the outside air is cooler, ventilation removes heat from the home.
- Internal heat generation accounts for heat produced by both humans and electrical appliances. We assume the heat output of a human to typically be around 100W. Our expression is then $100n_{ppl} + P_{app}$.

Combining all of these components results in the final differential equation shown below, which we walk through the process of solving in the next step.

$$\frac{dT}{dt} = -k_{con}(T_i - T_{env}) + a_{solar} I_{solar} + k_{vent}(T_i - T_{env}) + 100n_{ppl} + P_{app}.$$

1.3.4 Model Execution

To solve the differential equation we derived, we need data on all variables associated with heat capacity, conduction, radiation, and ventilation, as they vary for different housing types. Similarly, so does the power released by appliances. We outsourced the data from the tables below and used its values to calculate the temperature over the 24 hour period.

Housing Type and Relevant Variables (C, k_{con}, a_{solar}, I_{solar}, k_{vent})

Housing Type	C (MJ/K)	k_{con} (W/m²·K)	a_{total} (m²)	k_{vent} (W/K)
Modern Apartment	8.5	0.35	210	90
1980s Apartment	7.2	0.45	190	120
Older Brick Apartment	12.3	0.7	220	180
Modern Single-Family Home	15.7	0.3	380	120
1970s Single-Family Home	13.8	0.5	340	250
Pre-1950s Single-Family Home	10.5	0.75	290	300

Figure 2[3]. Relevant Conductivity, Ventilation Variables for Housing

Number of Occupants and Power of Appliances

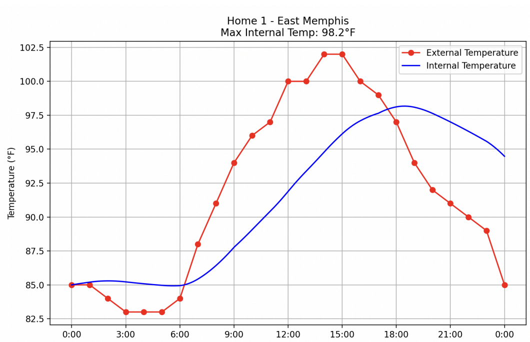
Occupancy Pattern	Occupants	P_{app} (W)
Working Family (Daytime Empty)	4	250 (morning/evening), 50 (daytime)
Retired Couple (Full Day)	2	150 (constant)
Single Occupant (Mixed)	1	100 (variable)
Work-From-Home Family	4	300 (daytime), 150 (nighttime)

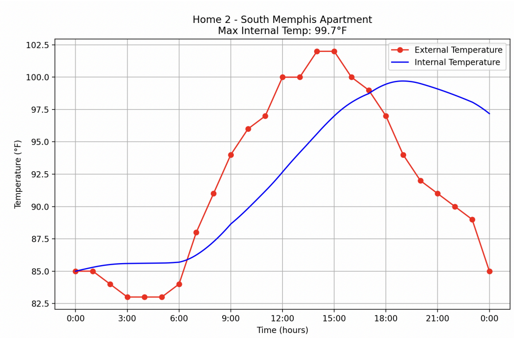
Figure 3[2]. Realistic Power Usage

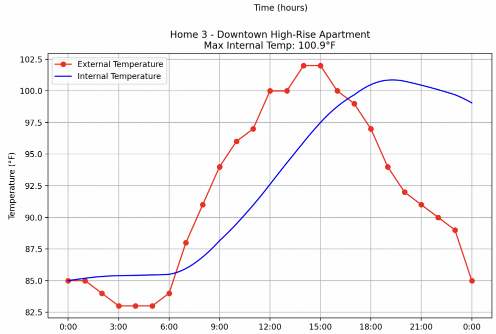
As shown in the table above, we use different values for the P_{app} values throughout the day to realistically represent appliance power consumption. To model the difference in the amount of heat due to the sun over the course of the day, we use the approximate solar radiation with a sinusoidal function. Regarding the computational implementation of this model, we wrote a python script to calculate $\frac{dT}{dt}$ over the total time period. The python program we wrote uses Euler's method to iteratively multiply numerically calculated values of $\frac{dT}{dt}$ by small steps t to derive the final function $T(t)$, which represents the final temperature over the 24 hour period.

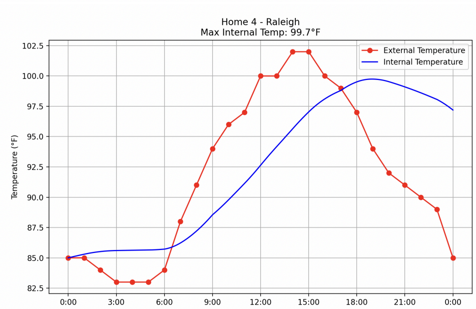
1.4 Results

Using the housing data for each home and parameters defined for specific homes in the previous tables, we predicted the temperature of each dwelling over a 24 hour period. As shown above, we used Python to compute the solution to the differential equation and plotted the predicted temperature alongside the external temperature.









Figures 4, 5, 6, 7 for Homes 1, 2, 3, 4 respectively

Below is a table representing the predicted temperature of each of the different homes in 3 hour intervals over the total time period.

Hour	External Temp.	Home 1 Temp. (°F)	Home 2 Temp. (°F)	Home 3 Temp. (°F)	Home 4 Temp. (°F)
0	85	85.00	85.00	85.00	85.00
3	83	85.22	85.59	85.39	85.60
6	84	84.94	85.69	85.50	85.72
9	94	87.76	88.64	88.16	88.55
12	100	91.88	92.66	92.61	92.63
15	102	96.13	97.00	97.49	97.03
18	97	98.12	99.46	100.49	99.51
21	91	97.01	99.09	100.45	99.10
24	85	94.47	97.18	99.052	97.19

Figure 8. Modeled Temperature

1.5 Discussion

Our model estimates that at 6 hours, the temperatures of each home will be 84.94, 85.69, 85.50, and 85.72 °F for House 1, House 2, House 3, and House 4 respectively. From the data table, we can see that the peak temperature experienced by each house occurs between 18 and 21 hours, with each house's highest temperature in three-hour intervals as 98.12, 99.46, 100.49, and 99.51 °F. Overall, the trends of the temperature changes in each house seem to be similar, with slight differences likely caused by disparities in factors like insulation, shade, and heat emittance internal to the home.

1.6 Sensitivity Analysis

For Home 4, we generated a graph that overlays the minimum and maximum temperatures in a home over time with 10% perturbations in each starting coefficient and the starting temperature. As shown in the figure below, the maximum temperatures for the range of possible curves for internal temperature proves less than a one degree difference at the peak and end temperatures. This demonstrates strong resilience to changes in initial conditions.

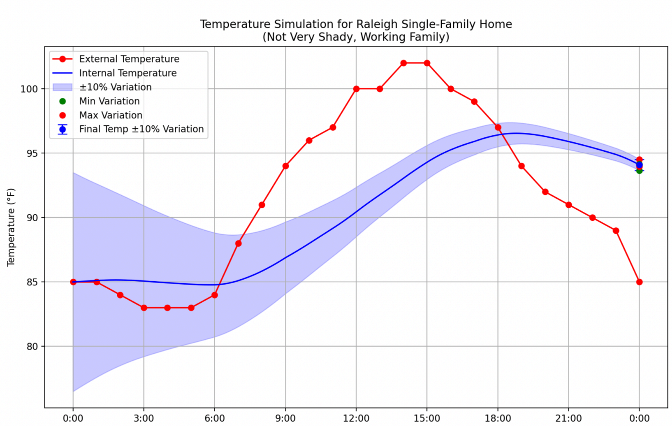


Figure 9. Model Sensitivity Analysis Graph

1.7 Strengths and Weaknesses

1.7.1 Strengths

Our differential equation model realistically captures the most significant factors that impact the temperature of a home experiencing heat waves over time, taking into account thermal conductivity, solar radiation, heat emittance from humans and appliances, and ventilation. Using Newton's Law of Heating as a foundation for the model aims to align our outcome with pragmatic physical principles, making the model more intuitive and grounded.. The model proved resilient to changes in initial conditions, as sensitivity analysis shows that small variations in starting temperature and coefficients have minimal impact on the final results.

1.7.2 Weaknesses

Our model, however, treats the home as a single homogeneous thermal zone, ignoring spatial temperature gradients caused by uneven insulation, multi-story layouts, or localized heat sources. While sensitivity analysis demonstrates resilience to minor parameter changes, the model may struggle with extreme deviations due to extreme weather changes or variables that may depend on one another, such as ventilation and conduction.

Q2: Power Hungry

2.1 Defining The Problem

The second problem asks us to model the peak power demand of the power grid, making predictions over the next 20 years for the city of Memphis, Tennessee. We used a multivariate regression with various independent factors to draw an accurate conclusion.

2.2 Assumptions

2.2.1.**The temperature readings taken in Memphis are considered representative of the whole city.**

- Justification: The city of Memphis has a relatively uniform climate and is geographically compact, meaning temperature variations across the city are minimal on a given day. Using a single temperature reading simplifies the model while still capturing major trends in heat wave conditions.

2.2.2. **The examined historical trends in the city will stay relatively constant over the next 20 years.**

- Justification: Many of the variables within our regression (ie socioeconomic status, population, GDP) exhibited steady trends over the past 10 years. We are assuming that the policies that Memphis takes remain historically comparable, allowing us to confidently model the direction that energy consumption seems to be heading. Furthermore, the model assumes that no unforeseen extreme events will occur during the predicted time frame. While heat waves themselves are certainly extreme, larger national crises would be both difficult to predict and could disrupt the underlying data our model relies on, affecting its accuracy and validity.

2.2.3. **All terms are linearly related.**

- Justification: Historical data indicates a strong linear correlation between indoor temperature changes and external factors such as outdoor temperature, time of day, and solar radiation.

Deviations from linearity, such as extreme temperature-dependent effects, are either rare or within a range where linear approximation remains statistically valid.

2.2.4. **No other independent variables will affect the peak power demand of Memphis.**

- Other independent factors like humidity and wind speed have a secondary influence and are often multicollinear with more significant variables. Thus, their presence would not only be statistically insignificant but also overcomplicate the model.

2.2.5. **Homoscedasticity is satisfied.**

- Variance in residuals remains consistent across different temperature levels in observed data, confirming that prediction errors do not systematically increase at higher or lower temperatures. This ensures the model remains statistically robust across all conditions analyzed.

2.2.6. **COVID-19 will have no effect on predicted energy consumption.**

- Post-2021 data indicates a return to pre-pandemic energy usage patterns. Utility reports and consumption trends in Memphis show that any temporary shifts in demand caused by lockdowns had largely stabilized, with no lasting structural changes in residential or commercial electricity consumption. Including pandemic-related adjustments would introduce unnecessary complexity without providing meaningful improvements in predictive accuracy.

2.3 Building the Model

2.3.1 Data Collection

After reviewing the given data statement, we decided that the best course of action was to find new data. Although relevant, Shelby County's annual electricity usage[1] lacked specificity to predict peak electricity consumption, while the monthly electricity consumption for East South Central USA was too broad[1]. Instead, we gathered annual data (2012-2022) from reputable sources such as Memphis Light, Gas, and Water (MLGW, the main electricity provider in Memphis).

Our selected predictors were peak temperature[14], population of Memphis[15], GDP[16], investment in the power grid[17], annual electricity consumption, rate charged for electricity in summer[18], and efficiency – measured as consumption divided by GDP to reflect energy use per dollar of economic output. The response variable, maximum power demand, was defined as the annual peak hourly load on the power grid[19], providing a more precise measure than consumption over a year given the prompt at hand.

2.3.2 Model Development

After collecting data, we evaluated relevant predictors to select a model. The models considered were a time series accounting for cyclic energy consumption, a logistic model, and a multivariate linear regression. Time series was ruled out due to limited data points and its inability to incorporate external predictors. Logistic regression was ruled out because there was no clear evidence of a "carrying capacity" for energy consumption, especially given Memphis's growing population. Ultimately, we chose multivariate linear regression to assess the correlation between predictors and peak demand.

We used Python to import all data and began constructing the multivariate linear regression model. We used the following equation, where Y is the annual peak hourly load on the power grid, ε is an error term, β_0 is a constant, β_i are coefficients of independent variables, and X_i are the values of the predictors:

$$Y = \beta_0 + \varepsilon + \sum_{i=1}^{n} \beta_i X_i$$

The model was calculated using Python's *LinearModels* library. The predictors used were determined using backwards selection, removing the predictor with the highest P-value, because a high P-value shows that the predictor is not statistically significant. This process was repeated until all remaining predictors had P-values below 0.1, indicating all predictors hold statistical significance. This model used the three predictor variables: peak temperature, GDP, and investment in the power grid. This model had an R-squared value of 0.704, showing significant explanatory power. However, to check for multicollinearity, a heat map showing correlations between each variable was created.

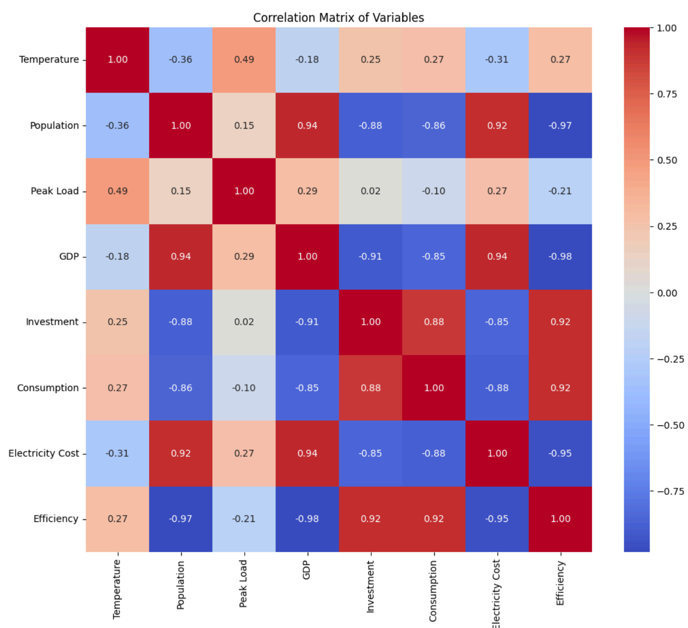


Figure 10. Heat map showing correlation between predictors

From the heat map, we found that investment and GDP were highly correlated with a coefficient of -0.91, suggesting multicollinearity. In order to account for this, we removed investment, with the resultant multivariate linear regression equation modeled by the following:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

Symbol	**Variable**	**Units**
Y	Annual Peak Hourly Load	MegaWatts (mW)
β_0	Constant Term	MegaWatts (mW)

β_1	Coefficient of Peak Temperature	MegaWatts/Degrees Fahrenheit ($\frac{mW}{F}$)
X_1	Peak Temperature	Degrees Fahrenheit (F)
β_2	Coefficient GDP	MegaWatts/Million Dollars ($\frac{mW}{\$}$)
X_2	GDP	Millions of Dollars ($)
ε	Error term	MegaWatts (mW)

Figure 11

2.3.3 Model Execution

We then forecasted the Peak Temperature and GDP up to 2045 based on a linear model.

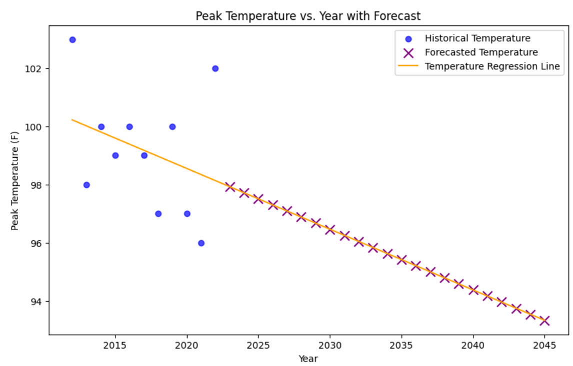


Figure 12. Peak temperature vs. year with forecast.

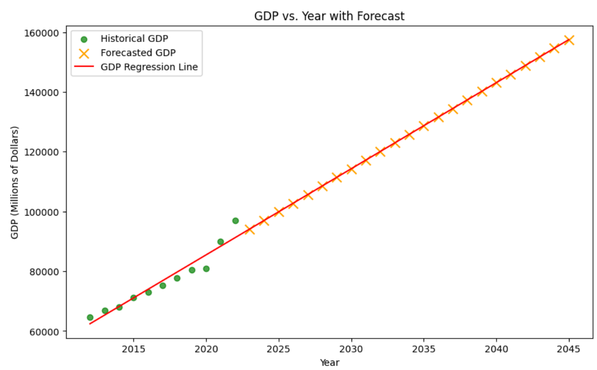


Figure 13. GDP vs. year with forecast.

Despite highly varied historical data, peak temperature clearly shows a downward trend, especially if two outliers in 2012 and 2022 are disregarded. GDP shows a clear, linear increase year after year.

Year	2045
Peak Temperature (Degrees Fahrenheit)	93.327
GDP (Millions of Dollars)	157466.226

Given this data, we can calculate estimates of Peak Electricity Demand in Memphis.

2.4 Results

We used a multivariate regression to model the peak power demand of the Memphis power grid over the next 20 years. The results are shown below:

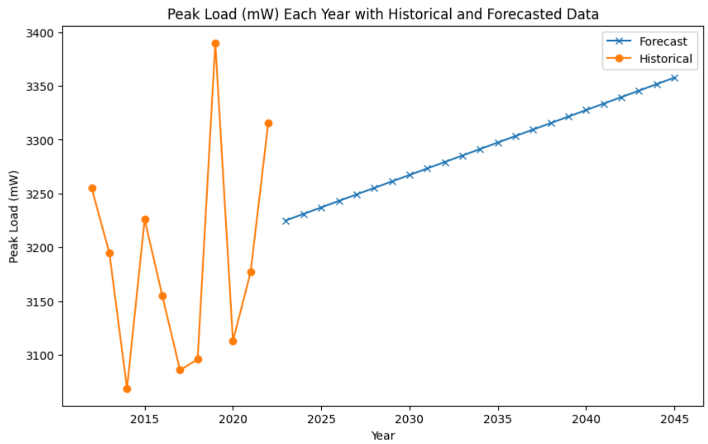


Figure 14. Peak load (mW) each year with both historical and forecasted data.
Forecasted Value for 2045 (20 years into the future): 3357.716 mW

2.5 Discussion

Ultimately, our multivariate regression model predicted that the peak load each year in Memphis would increase over time. Our forecasted peak electricity demand for 2045 was 3,357.72 mW, meaning that 20 years from now, given no unusual changes, Memphis's power grid will need to be able to handle at least 3,357.72 mW of power demanded in one hour. However, the increase is very slight, with a total of only 120.58 mW of change over these 20 forecasted years. This keeps all forecasted values within the range of the historical data, allowing us to conclude that there will not be any major changes in the maximum demand for power through these 20 years apart from the slight increase shown above.

2.6 Sensitivity and Error Analysis

In our multivariate regression analysis, we examined the residual plot, which displays residuals on the y-axis against predicted values (or sometimes individual independent variables) on the x-axis. The goal of this analysis is to identify whether any systematic patterns exist in the residuals, which would suggest that a linear model is insufficient to describe the relationship between variables.

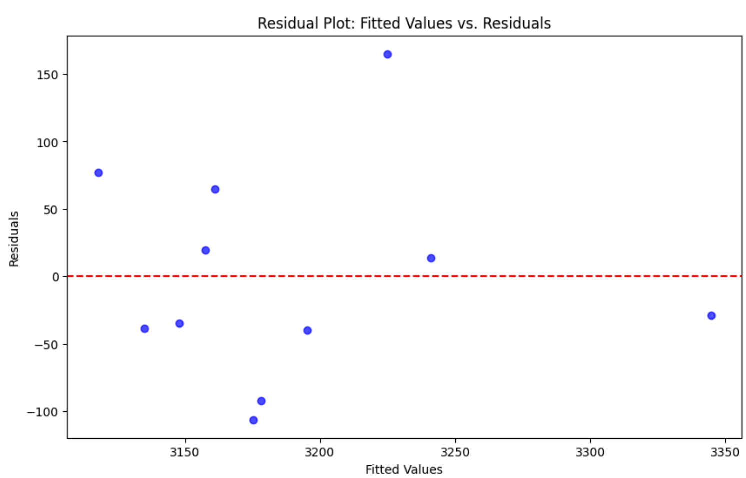


Figure 15. Residuals Between Predicted and Actual Peak Power Demand.

The residual plot for our model exhibits a random, widely spread distribution, with no discernible patterns or trends. This indicates that the relationship between the predictors and resultant variable is captured well by the linear model. Furthermore, it indicates that the residuals are independent, meaning that errors are not correlated with each other. Given these findings, we can confidently justify our decision to use a linear model for this regression. The residual plot supports the assumption that the relationship between our independent and dependent variables is well-approximated by a linear function.

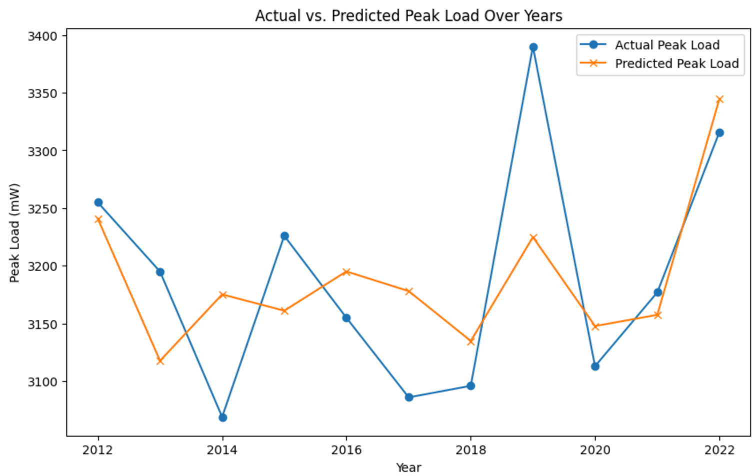


Figure 16. Actual vs Predicted Peak Load over Years

To determine the accuracy of our predictions, we plotted our model's predicted values for homeless people for the years 2012 to 2022 against the historical data. Then we calculated the Mean Absolute Percent Error (MAPE), to measure the accuracy of our model. The resulting MAPE was 1.94%. This low value indicates our model is highly accurate.

2.7 Strengths and Weaknesses

2.7.1 Strengths

One major strength of our model is its transparency and interpretability. Unlike black-box models, which rely on methods not easily understood, multivariate models, when developed with sufficient computing, provide direct and quantifiable insight of model fit through correlation matrices, p-values, and F-statistics. Furthermore, we were able to take two factors – peak temperature and GDP – into account. They were both correlated with peak power use while remaining independent of each other, which allowed us to achieve more accurate results than just relying on just one variable.

2.7.2 Weaknesses

However, the model had a few limitations. It assumes a linear relationship between predictors and the response variable, which may not always hold in complex real-world data. The regression is also highly sensitive to outliers, as extreme values can disproportionately influence the model and distort results. Furthermore, the training data, especially for peak temperature and peak load, were highly variable. This may contribute to our model's uncertainty. In future research, if we could find more data, it would be more helpful to train on peak load and temperature over a longer period, such as a month.

Q3: Beat the Heat

3.1 The Problem

The third problem asks us to construct a metric to quantify the vulnerability of the various neighborhoods in Memphis to heat waves and power outages. We used a Min-Max normalized vulnerability score with four factors to identify the most at-risk areas and guide resource allocation for heat wave preparedness and emergency response.

3.2 Assumptions

3.2.1. **Min-Max normalization is valid for the given data.**

- Justification: The model assumes that transforming each variable to a 0–1 scale (via Min-Max) is appropriate and that the resulting normalized values maintain meaningful comparisons across different ZIP codes. There are no extreme enough outliers to distort the 0–1 scaling to an extent that would misrepresent the vulnerability of most neighborhoods. Some factors may be skewed, concentrating scores among a smaller number of neighborhoods, which may indicate that people and developments more greatly affected by heat waves have concentrated in a few vulnerable areas. Because Min-Max Scaling is a linear transformation, normalized scores will preserve such skewed distributions.

3.2.2. **All citizens of Memphis behave like rational actors.**

- Justification: This assumption is justified by the principle that individuals act in ways that protect their health and well-being. Older adults, who often have heightened sensitivity to heat, limit their exposure by spending more time in air-conditioned or shaded environments. People who travel to work by walking or taking public transit may choose to drive instead, if the improved comfort during heatwaves outweighs the financial costs. By assuming predictable behavior by Memphians during heat events, we can more accurately model which groups will be more susceptible to its negative consequences.

3.2.3. **All chosen factors in our model affect the vulnerability score equally.**

- Justification: Each of the four selected factors—income, population, proportion of elderly residents, and reliance on walking and public transit—addresses a unique facet of heat vulnerability. Income influences access to cooling and healthcare, a large population puts strain on infrastructure, demographic composition shapes health risk profiles, and transport modes determine exposure levels. Assigning equal weights ensures that no single dimension overshadows the others, allowing all four factors to contribute comparably to overall vulnerability.

3.2.4. **Each zip code consists of a neighborhood.**

- Justification: ZIP codes serve as well-established administrative units that reliably encompass demographics, infrastructure, and economic conditions[19]. The population, physical environment, and resource availability within these boundaries collectively function as a neighborhood, allowing for meaningful comparisons across the city.

3.3 The Model

3.3.1 Defining the Model

The vulnerability of neighborhoods to heat waves depends on a number of factors[20, 21], the most significant being income, total population, elderly population per household, and number of walkers. To determine which factors should be considered in the development of the model, a correlation matrix was created using Python's Statsmodels library:

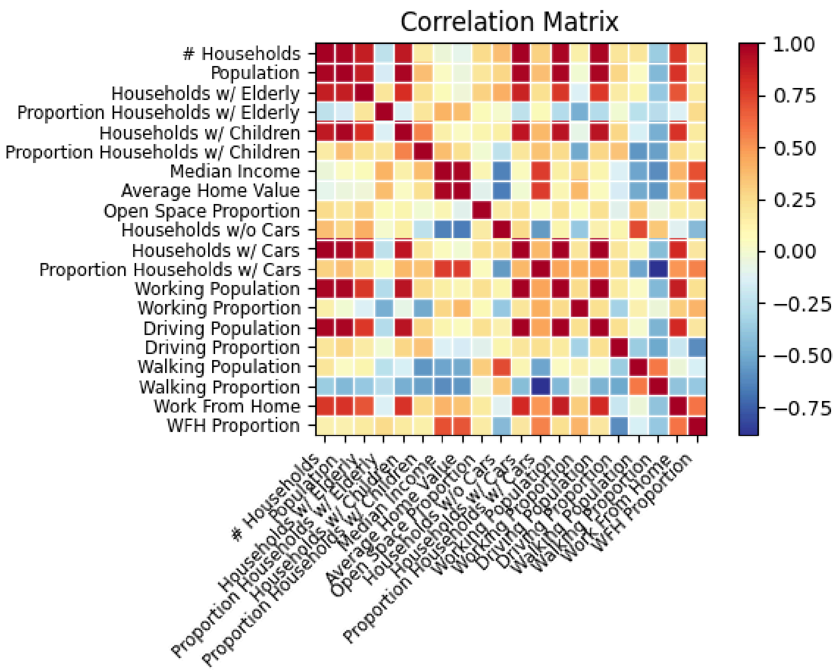


Fig. 17: Correlation matrix between various factors of ZIP codes in Memphis, Tennessee

1. **Income:** Households with lower incomes often lack access to air conditioning and reside in neighborhoods with fewer green spaces, more heat-trapping infrastructure, and less access to healthcare, all of which worsen the effects of heat exposure. The team chose to use each ZIP code's median household income to quantify the economic status of the neighborhood. Median income was chosen instead of other collinear factors, such as average home value or the employment-population ratio, because it provides a direct insight into the economic standing of a typical household in the neighborhood, instead of the properties or employment options in the area, which may vary based on land availability or type of industry.
2. **Population:** In neighborhoods with larger populations, more people are at risk of heat illnesses, and heat induced blackouts become more likely due to overloaded power grids. Population was found to be correlated with many other variables, including the number of households, the working population, and the number of detached whole houses; the team found that in many cases, this correlation could be accounted for by dividing a variable by the population size or number of households, e.g. considering the proportion of the population currently employed.
3. **Elderly Residents:** Older residents spend more time at home, placing greater strain on the municipal power grid, and have underlying health conditions that may be worsened by heat conditions or add to the overwhelming demand for local medical services. To avoid collinearity with population, the team considered the proportion of households containing 1 or more people over the age of 65.

4. **Walking and Public Transit:** ZIP codes whose residents are more likely to walk to work or use public transportation will be more severely affected by heat waves. These Memphis residents expose themselves to the extreme outside conditions as they walk or wait for their bus. Water and shade is often not available outside of their homes and workplaces. The raw number of people who walk to work shares collinearity with population; however, considering the proportion of Memphians who walk to work allowed the team to consider how big of a role walking and public transit play in each ZIP code.

Metrics for all four factors were normalized to range from 0 to 1 using Min-Max Scaling:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

A final vulnerability score, ranging from 0 to 4, was determined by performing a weighted sum of the four normalized scores. Because we assumed each of the four factors - economy, population, demographics, and transportation - would have equally weighted effects on a community's vulnerability, each of the four factors was given a weight of 1.

$$V = \sum_{f=1}^{4} w_f X'_f$$

3.4 Results

The model provided the following vulnerability scores for each of the given ZIP codes in Memphis:

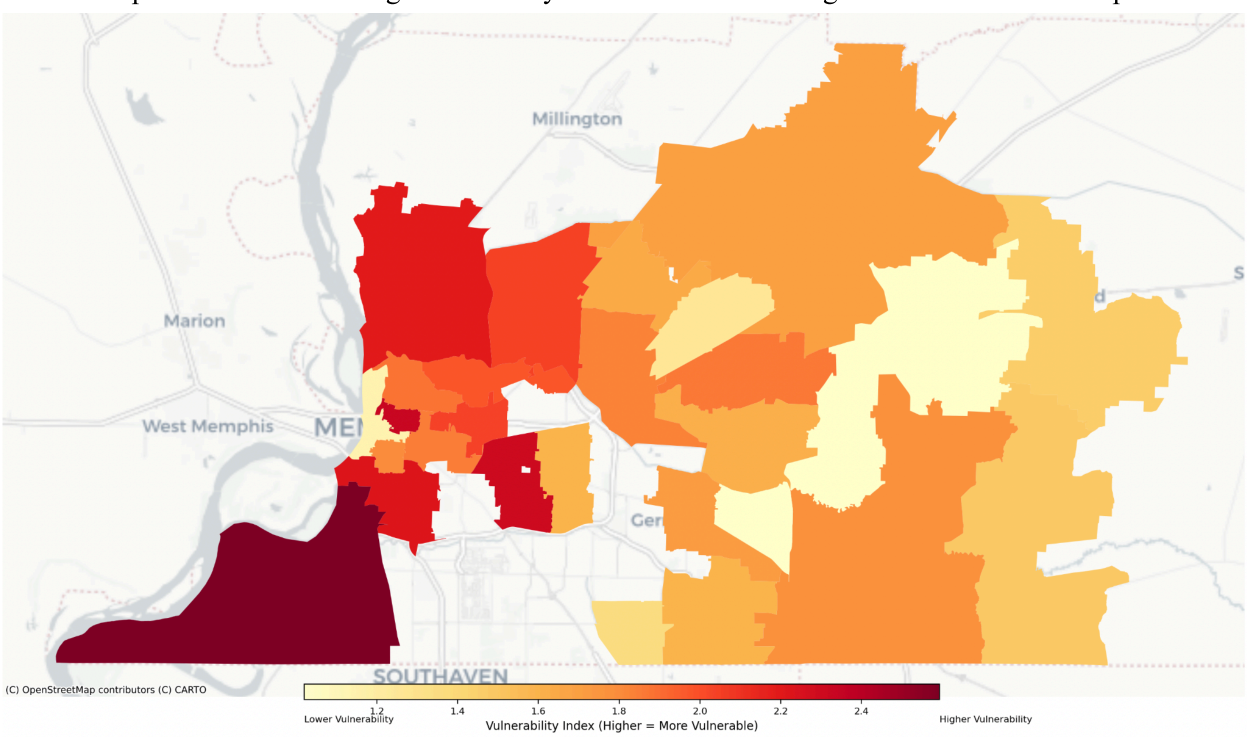


Fig. 18: Map showing vulnerability scores in selected Memphis ZIP codes regions

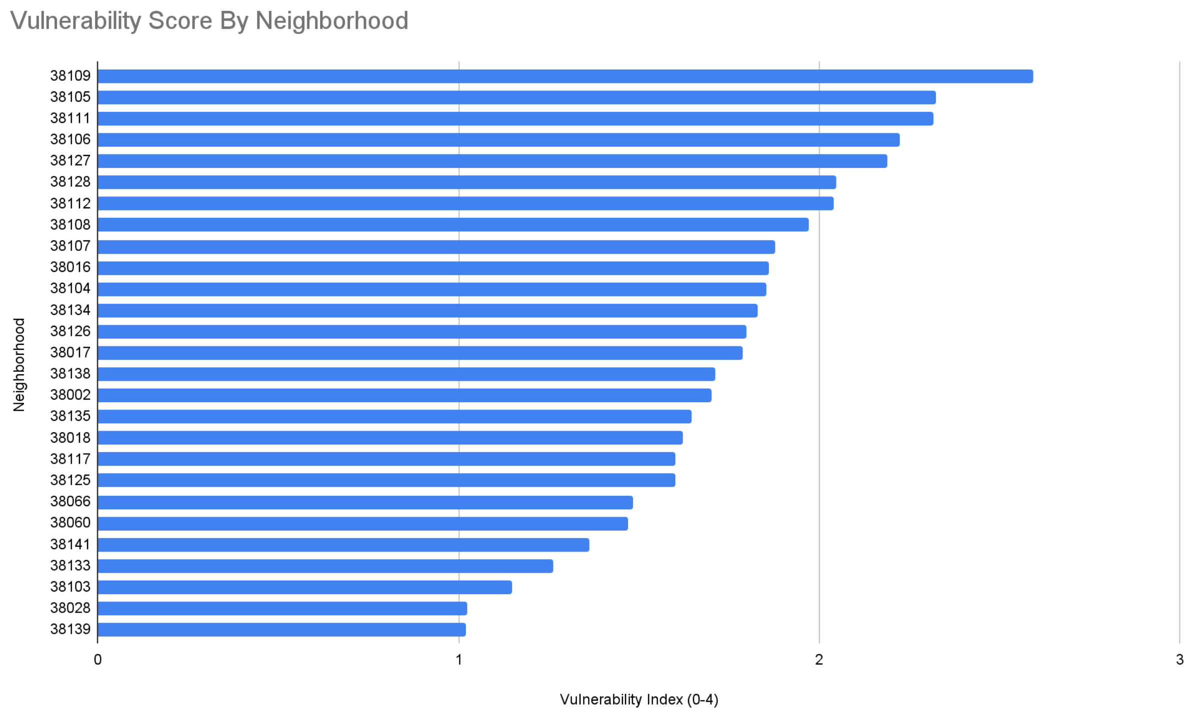


Fig. 19: Bar chart showing vulnerability scores in selected Memphis ZIP codes

3.5 Discussion

Our model calculates vulnerability scores for a variety of neighborhoods in Memphis, taking into consideration the effects of income, population, age demographics, and transportation. A weighted sum of the normalized scores for each factor was used to compute the vulnerability of each neighborhood. The resulting model quantifies heat-risk in a straightforward manner and can be generalized to any ZIP code in any location. The City of Memphis can use this scoring model to inform policy decisions, dedicating more of its resources to more vulnerable communities. In concentrating its efforts to reduce heat-risk in more neighborhoods that are more heavily impacted by extreme heat, the City can ensure a more equitable distribution of resources.

3.6 Policy Recommendation

The team recommends that the City of Memphis dedicate funding to a Cool Neighborhoods initiative, improving its streets to be more comfortable during periods of extreme heat. The city should rehabilitate streets and sidewalks in vulnerable neighborhoods with reflective, heat-reducing pavement and shade provided by trees and bus shelters. The community improvements provided by urban greenery and better bus stops, reducing heat, and pollution, and flooding from the nearby Mississippi river, will reduce the need for personal investment, thus lifting residents out of poverty[22]. Residents will be able to walk on surfaces made cooler by reflective pavement and a shaded canopy, without the discomfort or danger of extreme heat, and enjoy a safer and more comfortable wait for Memphis's notably low-frequency buses (the most frequent buses arrive every 30 minutes)[23]. The naturally cooler environment will reduce the need for air conditioning, and when seniors need to leave their homes to visit stores or social spaces, they will be less at risk of heat stroke. By dedicating municipal resources toward cooling down urban spaces, especially in high-risk neighborhoods, the city can protect vulnerable communities, limit exposure to extreme weather, and improve public health - without requiring individuals to spend money.

3.7 Strengths and Weaknesses

3.7.1 Strengths

The model assigned the highest vulnerability scores to neighborhoods with low-income residents, larger and more concentrated populations, high-density affordable housing, ports along the Mississippi river, hospitals, and major sources of employment - the people and places hit hardest by extreme heat and loss of power. Thus, the vulnerability index effectively identifies areas that face increased threats from heat waves and power outages, and provides the City of Memphis with information that will prove critical in allocating its resources to prepare for and respond to heat waves.

3.7.2 Weaknesses

The model had some drawbacks in how each factor was weighted and scaled. The team assumed that each of the four chosen factors were equally impactful on a community's heat risk. However, the factors may be different in terms of how they affected communities undergoing a heatwave. For example, the strain that the elderly put on medical and municipal services will likely be far less than the strain of a population's limited budget. Additionally, all factors were transformed linearly. This created the possibility of a normalized score distributed more asymmetrically. The proportion of the population that walked to work, for example, was heavily skewed toward a very small number of neighborhoods, because dense, walkable development is often concentrated near the city's downtown. Alternative normalization methods, such as mean normalization or z-score normalization, were considered; however, the team decided that concentrating high scores among a few key neighborhoods was an important part of the scoring process and reflected the tendency of vulnerable populations to concentrate near community centers.

Conclusion

Heat waves are becoming increasingly severe and frequent, posing significant risks to public health, infrastructure, and economic stability. This paper examined the dangers of extreme heat in Memphis, analyzing three key problems: indoor temperature, peak power demand forecasting, and vulnerability assessment.

Our temperature model revealed that non-air-conditioned dwellings could reach dangerously high internal temperatures of 97 to 99°F during a heatwave, regardless of their shading, floor size, or internal heat sources. This underscores the urgent need for cooling solutions, as lack of air conditioning remains the primary risk factor for extreme indoor heat.

Our analysis of future power demand used multivariate regression to forecast Memphis' peak electricity consumption through 2045. Despite rising temperatures, our model predicts only a small increase in maximum power demand (120.58 mW over 20 years), suggesting that existing power infrastructure may not require significant expansion to meet future heat-related energy needs.

Lastly, our vulnerability assessment analyzed 27 Memphis ZIP codes to determine which areas are most at risk during extreme heat events. By quantifying economic status, population density, age demographics, and transportation dependence, we identified neighborhoods such as South Memphis, the Medical District, and Downtown as being highly vulnerable. These areas, home to many low-income residents, will face disproportionate harm when power outages and extreme heat coincide, necessitating targeted mitigation efforts.

Summary Statistics

Q2:

Year	TemperatureF	TemperatureC	Population	Peak_Load_Hour_MW	GDP_Millions_Dollars	Investment_Thousands_Dollars	Consumption_KWH	Number_EV	Summer_ResidentialRate_Electricity_DollarPerKWH	Efficiency_KWH_Per_GDP
2012	103	39.4	1081000	3255	64569.542	977339	10753992000		0.06435	166548.9899
2013	98	36.7	1090000	3195	66932.313	996031	10705452000		0.066	159944.4501
2014	100	37.8	1100000	3069	68027.974	938844	10544122000		0.06771	154996.8547
2015	99	37.2	1109000	3226	71119.881	964812	10514853000		0.06928	147846.8869
2016	100	37.8	1119000	3155	72974.138	958213	10436626000		0.06644	143018.1471
2017	99	37.2	1129000	3086	75341.736	892669	10154668000		0.06804	134781.4444
2018	97	36.1	1139000	3096	77803.073	921736	10604732000		0.0714	136302.2255
2019	100	37.8	1144000	3390	80380.966	932380	10208674000		0.07416	127003.6242
2020	97	36.1	1150000	3113	81011.504	891294	9672364000	1179	0.07692	119394.9442
2021	96	35.6	1156000	3177	90039.953	841428	9800375000	1820	0.07788	108844.737
2022	102	38.9	1163000	3316	96976.712	844408	9768296000	3101	0.07937	100728.2656

```
=== 20-Year Forecast ===
    YearInt  Forecasted_TemperatureF  Forecasted_GDP  Forecasted_Peak_Load_MW
0      2023                97.927273    94112.605000              3225.082087
1      2024                97.718182    96992.315045              3231.110920
2      2025                97.509091    99872.025091              3237.139753
3      2026                97.300000   102751.735136              3243.168585
4      2027                97.090909   105631.445182              3249.197418
5      2028                96.881818   108511.155227              3255.226251
6      2029                96.672727   111390.865273              3261.255084
7      2030                96.463636   114270.575318              3267.283916
8      2031                96.254545   117150.285364              3273.312749
9      2032                96.045455   120029.995409              3279.341582
10     2033                95.836364   122909.705455              3285.370415
11     2034                95.627273   125789.415500              3291.399247
12     2035                95.418182   128669.125545              3297.428080
13     2036                95.209091   131548.835591              3303.456913
14     2037                95.000000   134428.545636              3309.485746
15     2038                94.790909   137308.255682              3315.514578
16     2039                94.581818   140187.965727              3321.543411
17     2040                94.372727   143067.675773              3327.572244
18     2041                94.163636   145947.385818              3333.601076
19     2042                93.954545   148827.095864              3339.629909
20     2043                93.745455   151706.805909              3345.658742
21     2044                93.536364   154586.515955              3351.687575
22     2045                93.327273   157466.226000              3357.716407
```

References

1. Hot Button Issue, MathWorks Math Modeling Challenge, curated data, https://m3challenge.siam.org/897bjhb54cgfc/.
2. https://static1.squarespace.com/static/5ff2adbe3fe4fe33db902812/t/6009dd9fa7bc363aa822d2c7/1611259312432/ISLR+Seventh+Printing.pdf

Q1:

3. https://www.nature.com/articles/s41598-023-39317-w
4. https://www.sciencedirect.com/topics/engineering/solar-peak-hour
5. https://archive.org/details/ASHRAEHANDBOOK1997
6. https://www.yourhome.gov.au/passive-design/thermal-mass
7. https://ergo.human.cornell.edu/studentdownloads/DEA3500notes/Thermal/thcondnotes.html
8. https://www.caee.utexas.edu/prof/Novoselac/classes/ARE346N/Handouts/ASHRAE_%20Gains_
9. https://archive.org/details/ASHRAEHANDBOOK1997

10. https://pmc.ncbi.nlm.nih.gov/articles/PMC10193453
11. https://www.yourhome.gov.au/passive-design/thermal-mass
12. https://ergo.human.cornell.edu/studentdownloads/DEA3500notes/Thermal/thcondnotes.html
13. https://www.caee.utexas.edu/prof/Novoselac/classes/ARE346N/Handouts/ASHRAE_%20Gains_Calculation_Tables.doc

Q2:

14. https://weatherspark.com/h/m/11921/2025/1/Historical-Weather-in-January-2025-in-Memphis-Tennessee-United-States#Figures-Temperature
15. https://www.macrotrends.net/global-metrics/cities/23063/memphis/population
16. https://www.abiresearch.com/news-resources/chart-data/ev-annual-power-consumption-per-vehicle#:~:text=Electric%20Vehicle%20Smart%20Charging%20Platforms&text=Electric%20Vehicles%20(EVs)%20demand%20a,kilowatt%20hours%20(KwH)%20annually.
17. https://www.mlgw.com/residential/residentialratesarchived
18. WE4YWI4Mzc5IiwidCI6IjlhZTI4OTlmLTU1ZWYtNDBhMC05NjEwLTZlZTkwZDNhMmFkMyJ9
19. https://www.mlgw.com/about/annualreport

Q3:

20. https://www.memphisrealtysearch.com/memphis-zip-codes/
21. https://www.census.gov/programs-surveys/acs
22. https://www.zillow.com/home-values/1388/shelby-county-tn/
23. https://treeboston.org/tree-equity-maps/benefits-of-urban-trees/
24. https://www.matatransit.com/assets/2/6/MATA-SysMap-Aug23-web.pdf

Code

Q1:

```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import os

def main():
    internal_temp = 85  # Initial internal temperature (F)

    # Get current directory where the script is located
    script_dir = os.path.dirname(os.path.abspath(__file__))
    csv_path = os.path.join(script_dir, 'data/heatwaves.csv')

    # Read the CSV file
    df = pd.read_csv(csv_path, skiprows=2)  # Skip the first two header rows

    # Extract time and temperature data
    times_str = df['Time'].values
    external_temps = df['Temperature (F)'].values

    # Convert time strings to hour values (0-24)
    hours = []
    for time_str in times_str:
        hour, minute = time_str.split(':')
        hour = int(hour)
        minute = int(minute.split()[0])  # Remove AM/PM

        # Convert to 24-hour format
        if 'PM' in time_str and hour != 12:
            hour += 12
        if 'AM' in time_str and hour == 12:
            hour = 0

        hours.append(hour + minute/60)

    # Ensure data covers full 24 hours
    if len(hours) > 0 and hours[-1] < 24:
        hours.append(24)
        external_temps = np.append(external_temps, external_temps[0])

    # Time points for simulation (granular) - full 24-hour period
    time_points = np.linspace(0, 24, 1000)

    # Housing type parameters
    housing_types = {
        "Modern Apartment": {"C_eff": 8.5e6, "u_value": 0.35, "U_vent": 90, "wall_area": 210, "ACH": 0.5},
        "1980s Apartment": {"C_eff": 7.2e6, "u_value": 0.45, "U_vent": 120, "wall_area": 190, "ACH": 0.8},
        "Older Brick Apartment": {"C_eff": 12.3e6, "u_value": 0.70, "U_vent": 180, "wall_area": 220, "ACH": 1.2},
        "Modern Single-Family Home": {"C_eff": 15.7e6, "u_value": 0.30, "U_vent": 120, "wall_area": 380, "ACH": 0.4},
        "1970s Single-Family Home": {"C_eff": 13.8e6, "u_value": 0.50, "U_vent": 250, "wall_area": 340, "ACH": 1.0},
        "Pre-1950s Single-Family Home": {"C_eff": 10.5e6, "u_value": 0.75, "U_vent": 300, "wall_area": 290, "ACH": 1.5},
        "East Memphis Single-Family Home": {"C_eff": 10.5e6, "u_value": 0.75, "U_vent": 300, "wall_area": 240, "ACH": 1.5},
        "South Memphis Apartment": {"C_eff": 7.2e6, "u_value": 0.45, "U_vent": 120, "wall_area": 180, "ACH": 0.8},
        "Downtown High-Rise Apartment": {"C_eff": 8.5e6, "u_value": 0.35, "U_vent": 90, "wall_area": 200, "ACH": 0.5},
        "Raleigh Single-Family Home": {"C_eff": 13.8e6, "u_value": 0.50, "U_vent": 250, "wall_area": 450, "ACH": 1.0}
    }
```

```python
# Define all four homes
homes = [
    {   # Home 1: East Memphis
        "name": "Home 1 - East Memphis",
        "housing_type": "East Memphis Single-Family Home",
        "solar_config": "Very Shady",
        "occupancy": "Family of 3",
        "custom_sq_ft": 950,
        "custom_u_value": 0.75,  # Pre-1950s (1953)
        "custom_wall_area": 240,
        "initial_temp": internal_temp
    },
    {   # Home 2: South Memphis Apartment
        "name": "Home 2 - South Memphis Apartment",
        "housing_type": "South Memphis Apartment",
        "solar_config": "Not Very Shady",
        "occupancy": "Family of 3",  # Updated to 3 occupants
        "custom_sq_ft": 675,
        "custom_u_value": 0.60,  # 1967 construction
        "custom_wall_area": 180,
        "initial_temp": internal_temp
    },
    {   # Home 3: Downtown High-Rise
        "name": "Home 3 - Downtown High-Rise Apartment",
        "housing_type": "Downtown High-Rise Apartment",
        "solar_config": "Not At All Shady",
        "occupancy": "Family of 2",
        "custom_sq_ft": 800,
        "custom_u_value": 0.35,  # Modern (2003)
        "custom_wall_area": 200,
        "initial_temp": internal_temp
    },
    {   # Home 4: Raleigh Single-Family Home
        "name": "Home 4 - Raleigh",
        "housing_type": "Raleigh Single-Family Home",
        "solar_config": "Not At All Shady",
        "occupancy": "Family of 6",
        "custom_sq_ft": 2993,
        "custom_u_value": 0.40,  # 1990 construction
        "custom_wall_area": 450,
        "initial_temp": internal_temp
    }
]

# Solar radiation estimation (W/m²) - simple model for a typical summer day
def solar_radiation(hour):
    if hour < 6 or hour > 20:  # Nighttime
        return 0
    else:  # Daytime with peak at noon
        return 800 * np.sin(np.pi * (hour - 6) / 14)
```

```python
# Solar gain parameters
solar_configs = {
    "Very Shady": {"alpha_solar": 0.5},
    "Not Very Shady": {"alpha_solar": 1},
    "Not At All Shady": {"alpha_solar": 2}
}


# Internal heat gain parameters (from the provided tables)
occupancy_patterns = {
    "Working Family": {
        "occupants": 4,
        "heat_per_person": 100,
        "appliance_load": {"morning": 250, "daytime": 50, "evening": 250, "night": 50}
    },
    "Retired Couple": {
        "occupants": 2,
        "heat_per_person": 90,
        "appliance_load": {"constant": 150}
    },
    "Single Occupant": {
        "occupants": 1,
        "heat_per_person": 100,
        "appliance_load": {"variable": 100}
    },
    "Work-From-Home Family": {
        "occupants": 4,
        "heat_per_person": 110,
        "appliance_load": {"daytime": 300, "nighttime": 150}
    },
    "Family of 3": {
        "occupants": 3,
        "heat_per_person": 100,
        "appliance_load": {"morning": 225, "daytime": 50, "evening": 225, "night": 50}
    },
    "Family of 2": {
        "occupants": 2,
        "heat_per_person": 100,
        "appliance_load": {"morning": 200, "daytime": 50, "evening": 200, "night": 50}
    },
    "Family of 6": {
        "occupants": 6,
        "heat_per_person": 100,
        "appliance_load": {"morning": 300, "daytime": 100, "evening": 350, "night": 75}
    }
}
```

```python
# Solar radiation estimation (W/m²) - simple model for a typical summer day
def solar_radiation(hour):
    if hour < 6 or hour > 20:  # Nighttime
        return 0
    else:  # Daytime with peak at noon
        return 800 * np.sin(np.pi * (hour - 6) / 14)

# Internal heat gain based on time of day
def internal_heat_gain(hour, occupancy):
    occupant_heat = occupancy["occupants"] * occupancy["heat_per_person"]

    if "Family" in selected_occupancy:
        # Morning: 6-9, Evening: 17-23, Night: 23-6, Daytime: 9-17
        if 6 <= hour < 9:  # Morning
            return occupant_heat + occupancy["appliance_load"]["morning"]
        elif 17 <= hour < 23:  # Evening
            return occupant_heat + occupancy["appliance_load"]["evening"]
        elif hour < 6 or hour >= 23:  # Night
            return occupancy["heat_per_person"] * occupancy["occupants"] + occupancy["appliance_load"]["night"]
        else:  # Daytime
            return occupancy["appliance_load"]["daytime"]  # Minimal presence

    elif "Retired Couple" in selected_occupancy:
        return occupant_heat + occupancy["appliance_load"]["constant"]

    elif "Work-From-Home" in selected_occupancy:
        if 8 <= hour < 22:  # Daytime
            return occupant_heat + occupancy["appliance_load"]["daytime"]
        else:  # Nighttime
            return occupancy["heat_per_person"] * (occupancy["occupants"]/2) + occupancy["appliance_load"]["nighttime"]

    else:  # Single occupant with variable pattern
        if 8 <= hour < 10 or 18 <= hour < 23:  # Active hours
            return occupant_heat + occupancy["appliance_load"]["variable"]
        else:  # Less active or away
            return occupancy["appliance_load"]["variable"] / 2

# Function to convert Fahrenheit to Celsius
def f_to_c(temp_f):
    return (temp_f - 32) * 5.0/9.0

# Create separate plots for all homes

# Initialize dict to store all simulation results
simulation_results = {
    'Hour': list(range(25)),
    'External_Temp': []
}

# Store the external temperatures first
for hour in range(25):
    simulation_results['External_Temp'].append(np.interp(hour, hours, external_temps))
```

```python
        # Store hourly temperatures (0-24 hours) from simulation results
        for hour in range(25):
            # Use interpolation to get exact hourly values
            simulation_results[f'{home_name}_Internal'].append(
                np.interp(hour, time_points, temperatures_f)
            )

        # Create plot for this home
        plt.figure(figsize=(10, 6))
        plt.plot(hours, external_temps, 'ro-', label='External Temperature')
        plt.plot(time_points, temperatures_f, 'b-', label='Internal Temperature')

        # Calculate max internal temperature
        max_internal_temp = round(np.max(temperatures_f), 1)

        # Title with details
        plt.title(f"{home['name']}\nMax Internal Temp: {max_internal_temp}°F")
        plt.xlabel("Time (hours)")
        plt.ylabel("Temperature (°F)")

        # Create time labels for x-axis
        time_labels = [f"{h%24}:00" for h in range(0, 25, 3)]
        time_ticks = np.arange(0, 25, 3)
        plt.xticks(time_ticks, time_labels)

        plt.grid(True)
        plt.legend()

        # Save each figure with a unique name
        plt.savefig(f'home_thermal_{i+1}_{home["name"].replace(" - ", "_").replace(" ", "_")}.png')

    # Create DataFrame from the simulation_results dictionary
    hourly_df = pd.DataFrame(simulation_results)

    # Make sure the directory exists
    os.makedirs(os.path.dirname(os.path.join(script_dir, 'data')), exist_ok=True)

    # Save to CSV
    csv_output_path = os.path.join(script_dir, 'data', 'hourly_temperatures.csv')
    hourly_df.to_csv(csv_output_path, index=False)

    print(f"Hourly temperatures saved to {csv_output_path}")

    # Show all figures
    plt.show()

if __name__ == "__main__":
    main()
```

Q2:

```python
# reading CopyofData.gsheet "Data Aggregation" for David

import gspread
import pandas as pd
from google.colab import auth
from google.auth.transport.requests import Request
from google.auth import default

# authenticate and get the credentials
auth.authenticate_user()
credentials, _ = default()

# authorize gspread with the credentials
gc = gspread.authorize(credentials)

# open the spreadsheet by its URL
spreadsheet = gc.open_by_url("https://docs.google.com/spreadsheets/d/1xZWrhGFlFFV6XSTQsBi38qKD7PdGNJOGkKy-NyxW91c/edit?usp=sharing")

# select the "Data Aggregation" sheet
worksheet = spreadsheet.worksheet("Data Aggregation")

# convert the sheet to a pandas DataFrame
data = pd.DataFrame(worksheet.get_all_records())
```

```
# Summary Statistics
cols_to_exclude = ['Year']
data.drop(columns=cols_to_exclude).describe().round(3)
```

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

#Heat Map

# Calculate the correlation matrix
correlation_matrix = correlation_data.corr()

# Define a dictionary mapping the original variable names to new names
rename_dict = {
   'TemperatureF': 'Temperature',
    'GDP_Millions_Dollars': 'GDP',
    'Investment_Thousands_Dollars': 'Investment',
    'Consumption_KWH': 'Consumption',
    'Population': 'Population',
    'Summer_ResidentialRate_Electricity_DollarPerKWH': 'Electricity Cost',
    'Efficiency_KWH_Per_GDP': 'Efficiency',
   'Peak_Load_Hour_MW': 'Peak Load',
    # Add additional mappings as needed for your data
}

# Rename both the columns and index labels of the correlation matrix
correlation_matrix.rename(columns=rename_dict, index=rename_dict, inplace=True)

# Plot the correlation matrix as a heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Variables')
plt.show()
```

```python
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt

# --- Read your data from the Google Sheet ---
spreadsheet = gc.open_by_url("https://docs.google.com/spreadsheets/d/1xZWrhGFlFFV6XSTQsBi38qKD7PdGNJOGkKy-NyxW91c/edit?usp=sharing")
worksheet = spreadsheet.worksheet("Data Aggregation")
data = pd.DataFrame(worksheet.get_all_records())

# Convert 'Year' column to datetime (assuming it's a 4-digit year)
data['Year'] = pd.to_datetime(data['Year'], format='%Y')

# --- Prepare dependent and independent variables ---
# y: dependent variable
y = data['Peak_Load_Hour_MW']

# X: independent variables (without any time dummies or panel specification), ran this multiple times for backward selection,
#until these three predictors were narrowed down
X = data[['TemperatureF', 'GDP_Millions_Dollars',
          'Investment_Thousands_Dollars']]#'Consumption_KWH','Population', 'Summer_ResidentialRate_Electricity_DollarPerKWH','Efficiency_KWH_Per_GDP'

# Add a constant to the independent variables for the intercept.
X = sm.add_constant(X)

# --- Fit the OLS regression model ---
model = sm.OLS(y, X)
results = model.fit(cov_type='HC1')  # using robust standard errors if desired

# Print the regression summary
print(results.summary())

# --- Create predictions ---
data['Predicted_Peak_Load'] = results.predict(X)
```

```python
# Optionally, display a plot of actual vs. predicted values over the years.
plt.figure(figsize=(10, 6))
plt.plot(data['Year'], data['Peak_Load_Hour_MW'], marker='o', label='Actual Peak Load')
plt.plot(data['Year'], data['Predicted_Peak_Load'], marker='x', label='Predicted Peak Load')
plt.xlabel('Year')
plt.ylabel('Peak_Load_Hour_MW')
plt.title('Actual vs. Predicted Peak Load Over Years')
plt.legend()
plt.show()
```

```python
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt

# --- Read your data from the Google Sheet ---
spreadsheet = gc.open_by_url("https://docs.google.com/spreadsheets/d/1xZWrhGFlFFV6XSTQsBi38qKD7PdGNJOGkKy-NyxW91c/edit?usp=sharing")
worksheet = spreadsheet.worksheet("Data Aggregation")
data = pd.DataFrame(worksheet.get_all_records())

# Convert 'Year' column to datetime (assuming it's a 4-digit year)
data['Year'] = pd.to_datetime(data['Year'], format='%Y')

# --- Prepare dependent and independent variables ---
y = data['Peak_Load_Hour_MW']
X = data[['TemperatureF', 'GDP_Millions_Dollars']]

# Add a constant to the independent variables for the intercept.
X = sm.add_constant(X)

# --- Fit the OLS regression model ---
model = sm.OLS(y, X)
results = model.fit(cov_type='HC1')  # using robust standard errors if desired

# Print the regression summary
print(results.summary())

# --- Create predictions ---
data['Predicted_Peak_Load'] = results.predict(X)

# --- Calculate % Error ---
data['Error'] = data['Peak_Load_Hour_MW'] - data['Predicted_Peak_Load']
data['% Error'] = (data['Error'] / data['Peak_Load_Hour_MW']) * 100  # Percentage error for each prediction
```

```python
# Calculate Mean Absolute Percentage Error (MAPE)
mape = data['% Error'].abs().mean()
print(f"Mean Absolute Percentage Error (MAPE): {mape:.2f}%")

# Optionally, display a plot of actual vs. predicted values over the years.
plt.figure(figsize=(10, 6))
plt.plot(data['Year'], data['Peak_Load_Hour_MW'], marker='o', label='Actual Peak Load')
plt.plot(data['Year'], data['Predicted_Peak_Load'], marker='x', label='Predicted Peak Load')
plt.xlabel('Year')
plt.ylabel('Peak Load (MW)')
plt.title('Actual vs. Predicted Peak Load Over Years')
plt.legend()
plt.show()

# Print individual errors
print(data[['Year', 'Peak_Load_Hour_MW', 'Predicted_Peak_Load', 'Error', '% Error']])
```

```python
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt

# ---------------------------------------------------------------------------
# 1. Load Data from Google Sheet
# ---------------------------------------------------------------------------
# Replace this with your actual code/credentials to authenticate and read the sheet
spreadsheet = gc.open_by_url("https://docs.google.com/spreadsheets/d/1xZWrhGFlFFV6XSTQsBi38qKD7PdGNJOGkKy-NyxW91c/edit?usp=sharing")
worksheet = spreadsheet.worksheet("Data Aggregation")
data = pd.DataFrame(worksheet.get_all_records())

# ---------------------------------------------------------------------------
# 2. Clean & Prepare Data
# ---------------------------------------------------------------------------
# Convert 'Year' to datetime, then create an integer year column
data['Year'] = pd.to_datetime(data['Year'], format='%Y')
data['YearInt'] = data['Year'].dt.year

# Keep only the columns we need
df = data[['YearInt', 'Peak_Load_Hour_MW', 'TemperatureF', 'GDP_Millions_Dollars']].copy()

# Drop any rows with missing values in these columns
df.dropna(inplace=True)

# ---------------------------------------------------------------------------
# 3. Fit Separate Time-Trend Models (Linear Regressions) for Temperature & GDP
# ---------------------------------------------------------------------------
# A) Temperature ~ YearInt
X_temp = sm.add_constant(df[['YearInt']])  # Expected columns: [const, YearInt]
y_temp = df['TemperatureF']
temp_model = sm.OLS(y_temp, X_temp).fit()
```

```
# B) GDP ~ YearInt
X_gdp = sm.add_constant(df[['YearInt']])   # Expected columns: [const, YearInt]
y_gdp = df['GDP_Millions_Dollars']
gdp_model = sm.OLS(y_gdp, X_gdp).fit()

# ------------------------------------------------------------------------------
# 4. Fit Main Linear Regression: Peak Load ~ Temperature + GDP
# ------------------------------------------------------------------------------
X_main = df[['TemperatureF', 'GDP_Millions_Dollars']]
X_main = sm.add_constant(X_main)  # Expected columns: [const, TemperatureF, GDP_Millions_Dollars]
y_main = df['Peak_Load_Hour_MW']
peak_model = sm.OLS(y_main, X_main).fit()

# Print summaries for clarity
print("\n=== Temperature ~ YearInt (Linear Regression) ===")
print(temp_model.summary())
print("\n=== GDP ~ YearInt (Linear Regression) ===")
print(gdp_model.summary())
print("\n=== Peak Load ~ Temperature + GDP (Linear Regression) ===")
print(peak_model.summary())

# ------------------------------------------------------------------------------
# 5. Forecast Next 20 Years
# ------------------------------------------------------------------------------
last_year = df['YearInt'].max()
future_years = range(last_year + 1, last_year + 24)  # e.g., if last_year=2022, then years 2023 to 2042

forecasts = []
for yr in future_years:
    # (A) Predict Temperature for future year
    # Manually build a DataFrame with a constant column and YearInt
    future_temp_df = pd.DataFrame({'YearInt': [yr]})
    future_temp_df['const'] = 1
```

```
forecasts = []
for yr in future_years:
    # (A) Predict Temperature for future year
    # Manually build a DataFrame with a constant column and YearInt
    future_temp_df = pd.DataFrame({'YearInt': [yr]})
    future_temp_df['const'] = 1
    # Ensure correct column order: const first, then YearInt
    future_temp_df = future_temp_df[['const', 'YearInt']]
    pred_temp = temp_model.predict(future_temp_df).iloc[0]

    # (B) Predict GDP for future year
    future_gdp_df = pd.DataFrame({'YearInt': [yr]})
    future_gdp_df['const'] = 1
    future_gdp_df = future_gdp_df[['const', 'YearInt']]
    pred_gdp = gdp_model.predict(future_gdp_df).iloc[0]

    # (C) Predict Peak Load using forecasted Temperature & GDP
    future_peak_df = pd.DataFrame({
        'TemperatureF': [pred_temp],
        'GDP_Millions_Dollars': [pred_gdp]
    })
    future_peak_df['const'] = 1
    # Ensure the column order matches that of X_main: [const, TemperatureF, GDP_Millions_Dollars]
    future_peak_df = future_peak_df[['const', 'TemperatureF', 'GDP_Millions_Dollars']]
    pred_peak_load = peak_model.predict(future_peak_df).iloc[0]

    forecasts.append({
        'YearInt': yr,
        'Forecasted_TemperatureF': pred_temp,
        'Forecasted_GDP': pred_gdp,
        'Forecasted_Peak_Load_MW': pred_peak_load
    })
```

```
# ---------------------------------------------------------------------------
# 6. Plot Historical vs. Forecasted Peak Load (Optional)
# ---------------------------------------------------------------------------
hist_plot_df = df[['YearInt', 'Peak_Load_Hour_MW']].copy()
hist_plot_df['Type'] = 'Historical'

fut_plot_df = forecast_df[['YearInt', 'Forecasted_Peak_Load_MW']].rename(
    columns={'Forecasted_Peak_Load_MW': 'Peak_Load_Hour_MW'}
)
fut_plot_df['Type'] = 'Forecast'

plot_df = pd.concat([hist_plot_df, fut_plot_df], ignore_index=True).sort_values('YearInt')

plt.figure(figsize=(10, 6))
for ttype, subset in plot_df.groupby('Type'):
    plt.plot(subset['YearInt'], subset['Peak_Load_Hour_MW'],
             marker='o' if ttype=='Historical' else 'x',
             label=ttype)

plt.xlabel('Year')
plt.ylabel('Peak Load (mW)')
plt.title('Peak Load (mW) Each Year with Historical and Forecasted Data')
plt.legend()
plt.show()

# Print forecast table
print("\n=== 20-Year Forecast ===")
print(forecast_df)

# Calculate fitted values and residuals from the main model
df['Fitted'] = peak_model.fittedvalues
df['Residuals'] = peak_model.resid
```


```
# Calculate fitted values and residuals from the main model
df['Fitted'] = peak_model.fittedvalues
df['Residuals'] = peak_model.resid
```


```
# Create a residual plot: Fitted Values vs. Residuals
plt.figure(figsize=(10, 6))
plt.scatter(df['Fitted'], df['Residuals'], color='blue', alpha=0.7)
plt.axhline(y=0, color='red', linestyle='--')
plt.xlabel('Fitted Values')
plt.ylabel('Residuals')
plt.title('Residual Plot: Fitted Values vs. Residuals')
plt.show()
```

```python
# ------------------------------------------------------------------------------
# GDP vs. Year with Forecasted Data
# ------------------------------------------------------------------------------
plt.figure(figsize=(10, 6))

# Plot historical GDP data
plt.scatter(df['YearInt'], df['GDP_Millions_Dollars'], color='green', alpha=0.7, label='Historical GDP')

# Plot forecasted GDP data
plt.scatter(forecast_df['YearInt'], forecast_df['Forecasted_GDP'], color='orange', marker='x', s=100, label='Forecasted GDP')

# Create an extended range of years from the minimum historical year to the maximum forecast year
years_extended = range(df['YearInt'].min(), forecast_df['YearInt'].max() + 1)
X_extended_gdp = sm.add_constant(pd.DataFrame({'YearInt': list(years_extended)}))
gdp_line = gdp_model.predict(X_extended_gdp)

# Plot the regression line over the extended period
plt.plot(list(years_extended), gdp_line, color='red', label='GDP Regression Line')

plt.xlabel('Year')
plt.ylabel('GDP (Millions of Dollars)')
plt.title('GDP vs. Year with Forecast')
plt.legend()
plt.show()
```

```python
# ------------------------------------------------------------------------------
# Temperature vs. Year with Forecasted Data
# ------------------------------------------------------------------------------
plt.figure(figsize=(10, 6))

# Plot historical Temperature data
plt.scatter(df['YearInt'], df['TemperatureF'], color='blue', alpha=0.7, label='Historical Temperature')

# Plot forecasted Temperature data
plt.scatter(forecast_df['YearInt'], forecast_df['Forecasted_TemperatureF'], color='purple', marker='x', s=100, label='Forecasted Temperature')

# Create an extended range of years from the minimum historical year to the maximum forecast year
years_extended = range(df['YearInt'].min(), forecast_df['YearInt'].max() + 1)
X_extended_temp = sm.add_constant(pd.DataFrame({'YearInt': list(years_extended)}))
temp_line = temp_model.predict(X_extended_temp)

# Plot the regression line over the extended period
plt.plot(list(years_extended), temp_line, color='orange', label='Temperature Regression Line')

plt.xlabel('Year')
plt.ylabel('Peak Temperature (F)')
plt.title('Peak Temperature vs. Year with Forecast')
plt.legend()
plt.show()
```

Q3:

```
#import necessary packages
import warnings
warnings.filterwarnings('ignore')

import statsmodels.api as sm
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import preprocessing
from sklearn import metrics
from statsmodels.stats.outliers_influence import variance_inflation_factor, OLSInfluence
#from sklearn.cross_validation import train_test_split

%matplotlib inline
```

```
[107] # Load data
      df = pd.read_csv('/Memphis_Zipcodes.csv')
      print(df.head())
```

```
[108] #clean up matrix by removing all unnecessary dependent variables
      df=df.drop('Neighborhood',axis=1).drop('Avg Value',axis=1).drop('Households w/ Bachelor\'s or Higher',axis=1).drop('ZIP code',axis=1)
```

```
[109] # lets plot correlation matrix using statmodels graphics packages's plot_corr

      # create correlation matrix
      corr = df.corr()
      sm.graphics.plot_corr(corr, xnames=list(corr.columns))
      plt.show()

      print(corr)
```

```python
import pandas as pd
import numpy as np

# Load the memzipcodes data
try:
    df = pd.read_csv('/Users/jackvu/Desktop/latex projects/compsci/m3mathworks/real/data/memzipcodesdata.csv')
except FileNotFoundError:
    print("File not found. Please provide the correct path to memzipcodes data.")
    exit(1)

# Function to normalize a column to range [0, 1]
def normalize_column(column):
    min_val = column.min()
    max_val = column.max()

    # Check if min and max are the same to avoid division by zero
    if min_val == max_val:
        return np.zeros(len(column))

    return (column - min_val) / (max_val - min_val)

# Store original zip codes
zipcodes = df['ZIP code'].copy() if 'ZIP code' in df.columns else df.index.copy()

# Calculate Walkers Ratio
df['Walkers Ratio'] = df['Walkers'] / df['Population']

# Extract only the columns we need
selected_columns = ['Population', 'Old Proportion', 'Median Income', 'Walkers Ratio']
filtered_df = df[selected_columns].copy()

# Normalize each column
normalized_df = pd.DataFrame()
for col in filtered_df.columns:
    normalized_df[col] = normalize_column(filtered_df[col])

# Reverse the Median Income score since we want lower incomes to contribute more positively
normalized_df['Median Income'] = 1 - normalized_df['Median Income']

# Calculate the sum of normalized scores for each zipcode
normalized_df['total_normalized_score'] = normalized_df.sum(axis=1)

# Add zipcode back to the dataframe
normalized_df['ZIP code'] = zipcodes

# Reorder columns to have zipcode first
cols = ['ZIP code'] + [col for col in normalized_df.columns if col != 'ZIP code']
normalized_df = normalized_df[cols]

# Display the results
print("Normalized data with total scores:")
print(normalized_df.head())

# Save to CSV
normalized_df.to_csv('normalized_memzipcodes.csv', index=False)
print("Saved normalized data to 'normalized_memzipcodes.csv'")
```

```python
import pandas as pd
import geopandas as gpd
import contextily as ctx
import matplotlib.pyplot as plt
import matplotlib.colors as colors

# Load the data with vulnerability scores
data = pd.read_csv('/Users/jackvu/Desktop/latex projects/compsci/m3mathworks/real/normalized_memzipcodes.csv')

# Load ZIP code geometry data from a local shapefile or use geopandas sample data
try:
    zipcode_gdf = gpd.read_file('/Users/jackvu/Desktop/latex projects/compsci/m3mathworks/real/data/tl_2022_us_zcta520.shp')
except:
    # Fallback to using the Memphis area ZIP codes only
    print("Local file not found. Downloading from Census Bureau TIGER/Line...")

    # Use the 2020 TIGER/Line Shapefiles which are commonly available
    url = "https://www2.census.gov/geo/tiger/TIGER2020/ZCTA520/tl_2020_us_zcta520.zip"
    try:
        # Download directly using geopandas
        zipcode_gdf = gpd.read_file(url)
    except Exception as e:
        print(f"Error downloading data: {e}")
        # Alternative source if Census data fails
        print("Using Natural Earth data as fallback...")
        zipcode_gdf = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
        # This would only give country boundaries, so we'd need to warn the user
        print("WARNING: Could not load ZIP code data. Map will show country boundaries only.")

# Convert ZIP code strings to match format in data
zipcode_gdf['ZCTA5CE20'] = zipcode_gdf['ZCTA5CE20'].astype(str)
data['ZIP code'] = data['ZIP code'].astype(str)

# Filter for Memphis area ZIP codes (these are typically in the range 38101-38199)
memphis_zip_codes = data['ZIP code'].unique()
zipcode_gdf_memphis = zipcode_gdf[zipcode_gdf['ZCTA5CE20'].isin(memphis_zip_codes)]
# Print info about the found ZIP codes
print(f"Found {len(zipcode_gdf_memphis)} Memphis area ZIP codes out of {len(memphis_zip_codes)} in the dataset.")

# Configure matplotlib colormap settings for better visualization
plt.rcParams.update({
    'axes.labelsize': 10,
    'legend.fontsize': 8,
    'legend.title_fontsize': 10,
})
# Set custom color scheme and legend label
cmap = 'YlOrRd'  # Yellow-Orange-Red colormap
legend_label = 'Vulnerability Index'

# If filtering doesn't return enough results, we can use a broader approach
if len(zipcode_gdf_memphis) < len(memphis_zip_codes) * 0.5:  # Less than half of expected ZIP codes
    print(f"Only found {len(zipcode_gdf_memphis)} of {len(memphis_zip_codes)} Memphis ZIP codes. Using broader geographic filter.")
    # Get ZIP codes for Tennessee (first 3 digits typically 370-385)
    tn_mask = zipcode_gdf['ZCTA5CE20'].str.startswith(('37', '38'))
    zipcode_gdf_memphis = zipcode_gdf[tn_mask]

# Merge the vulnerability data with geographic data
merged_data = zipcode_gdf_memphis.merge(data, left_on='ZCTA5CE20', right_on='ZIP code', how='left')
```

```python
# Convert to a proper CRS for mapping (Web Mercator)
merged_data = merged_data.to_crs(epsg=3857)

# Create the plot
fig, ax = plt.subplots(1, 1, figsize=(15, 10))

# Check if the column exists, otherwise use the first numeric column
if 'total_normalized_score' in merged_data.columns:
    score_column = 'total_normalized_score'
else:
    numeric_columns = merged_data.select_dtypes(include=['float64', 'int64']).columns
    score_column = numeric_columns[0] if len(numeric_columns) > 0 else merged_data.columns[0]
    print(f"Column 'total_normalized_score' not found. Using '{score_column}' instead.")

# Plot the data with a colormap indicating vulnerability
merged_data.plot(
    column=score_column,
    ax=ax,
    legend=False,  # Disable the default legend
    cmap='YlOrRd'
)

# Try to add basemap if contextily is available
try:
    ctx.add_basemap(ax, source=ctx.providers.CartoDB.Positron)
except:
    print("Could not add basemap - continuing without it")

# Add title and labels
plt.title('ZIP Code Vulnerability Heatmap', fontsize=12)
plt.axis('off')

# Create a separate axis for the colorbar
cbar_ax = fig.add_axes([0.25, 0.08, 0.5, 0.02])  # Adjusted position and size

# Create a custom colormap with normalized data range
norm = colors.Normalize(vmin=merged_data[score_column].min(), vmax=merged_data[score_column].max())
sm = plt.cm.ScalarMappable(cmap=cmap, norm=norm)
sm.set_array([])

# Add colorbar with descriptive labels
cbar = fig.colorbar(sm, cax=cbar_ax, orientation='horizontal')
cbar.set_label(f'{legend_label} (Higher = More Vulnerable)', fontsize=10)
cbar.ax.tick_params(labelsize=8)
# Add annotations with smaller font size and adjusted positions
ax.annotate('Lower Vulnerability', xy=(0.25, 0.05), xycoords='figure fraction', fontsize=8)
ax.annotate('Higher Vulnerability', xy=(0.75, 0.05), xycoords='figure fraction', fontsize=8)

plt.tight_layout()
plt.savefig('vulnerability_heatmap.png', dpi=300, bbox_inches='tight')
plt.show()
```