Homelessness in the United States (2007–2023): A Statistical Exploration

Introduction

The research examines both macroeconomic and individual factors that lead to homelessness across the United States during the 2007 to 2023 period. The study uses advanced statistical and machine learning methods combined with many socioeconomic indicators to discover the main factors driving shifts in homelessness throughout the years. The report analyzes public information from several sources including the U.S. Census Bureau and the Bureau of Labor Statistics and national health and housing surveys. Our analysis investigates linear and nonlinear relationships together with possible threshold effects using linear regression, Lasso, Double Machine Learning (DML), and Generalized Additive Models (GAM) beyond what traditional modeling techniques can detect. These methods reveal more than associations because they help uncover causal pathways while assisting with variable selection in high-dimensional data and describing predictor-response function shapes.

Data Description

The main dataset merges national time series information from 2007 through 2023 across multiple socioeconomic measures.

- PIT Homelessness Counts
- Unemployment Rate
- Consumer Price Index (CPI)
- Poverty Rate
- Median Household Income
- Gini Index (income inequality)
- Median Sales Price of Houses
- Rent CPI
- Illicit Drug Use (%)
- Depression Rate (%)
- Uninsured Rate (%)

Public government sources provided these variables which are then stored inside `homelessness_data.csv`. The dataset contains yearly national data for the United States in every observation. The dataset integrates structural and behavioral predictors to provide insights into homelessness dynamics from both economic and public health perspectives.

In [133…

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

from sklearn.linear_model import LinearRegression, LassoCV
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
from sklearn.preprocessing import StandardScaler, PolynomialFeatures
from sklearn.model_selection import cross_val_score, train_test_split
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score, mean_squared_error


from statsmodels.api import OLS, add_constant
from statsmodels.graphics.gofplots import qqplot
from statsmodels.stats.outliers_influence import summary_table

from econml.dml import LinearDML

from pygam import LinearGAM, s


import warnings
warnings.filterwarnings('ignore')


# Load the dataset
file_path = r"C:\Users\matia\Downloads\homelessness_data.csv"
df = pd.read_csv(file_path)
```

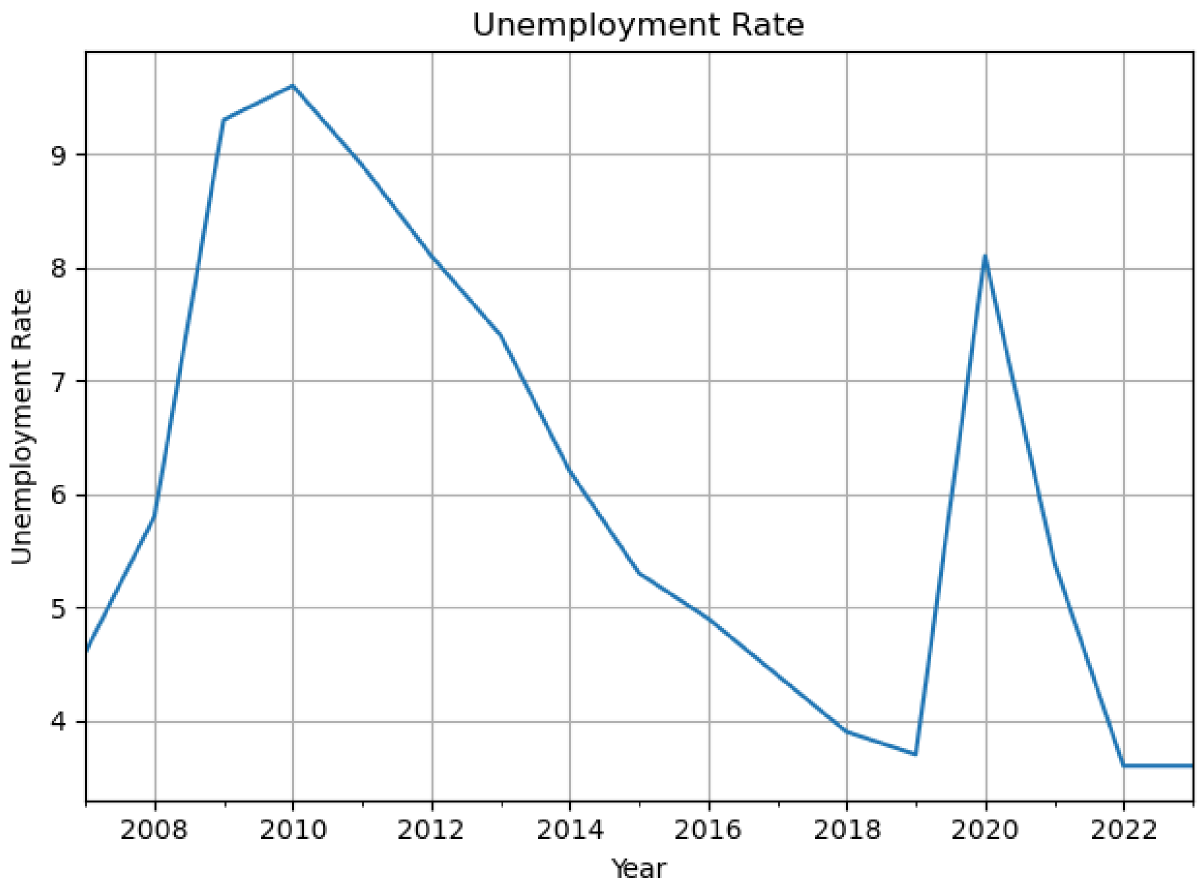
Exploratory Data Analysis

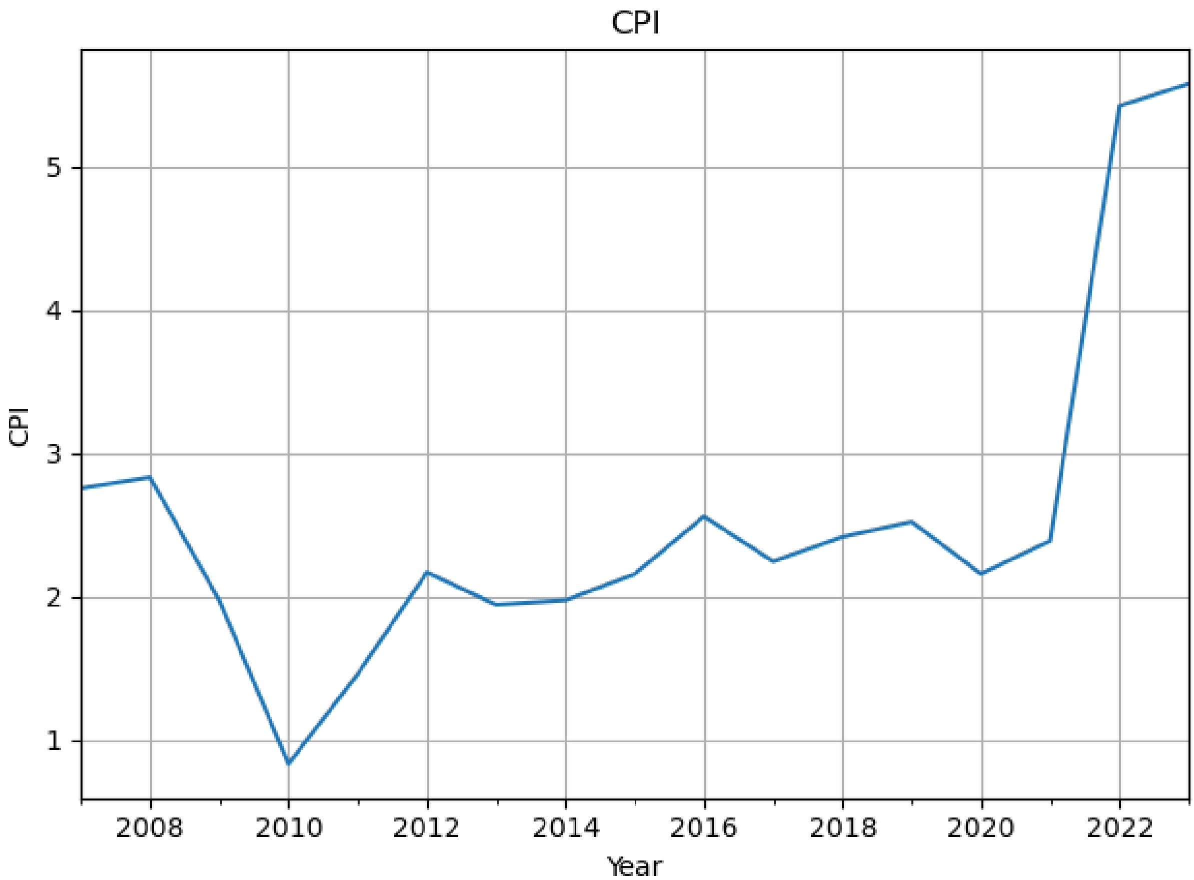
The time-series visualizations demonstrated diverging trends among predictors with variables like Uninsured Rate and Depression Rate showing stronger alignment with PIT Homelessness Counts than other variables. The correlation matrix measured these relationships revealing moderate to strong connections specifically between homelessness and health-related metrics. This statistical phase determines potential predictors while shaping model specifications. The research demonstrates homelessness as a multifaceted issue and supports using a multivariate analysis approach.

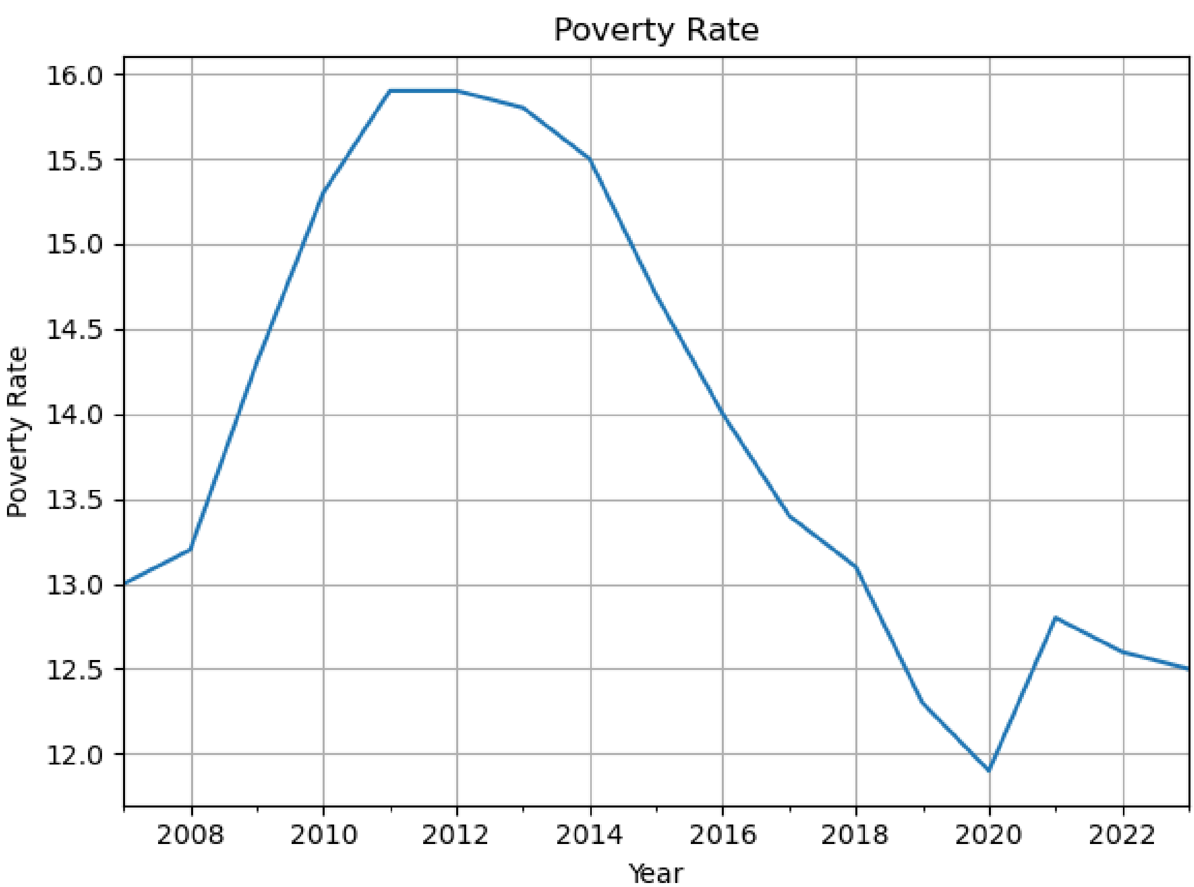
In [115…

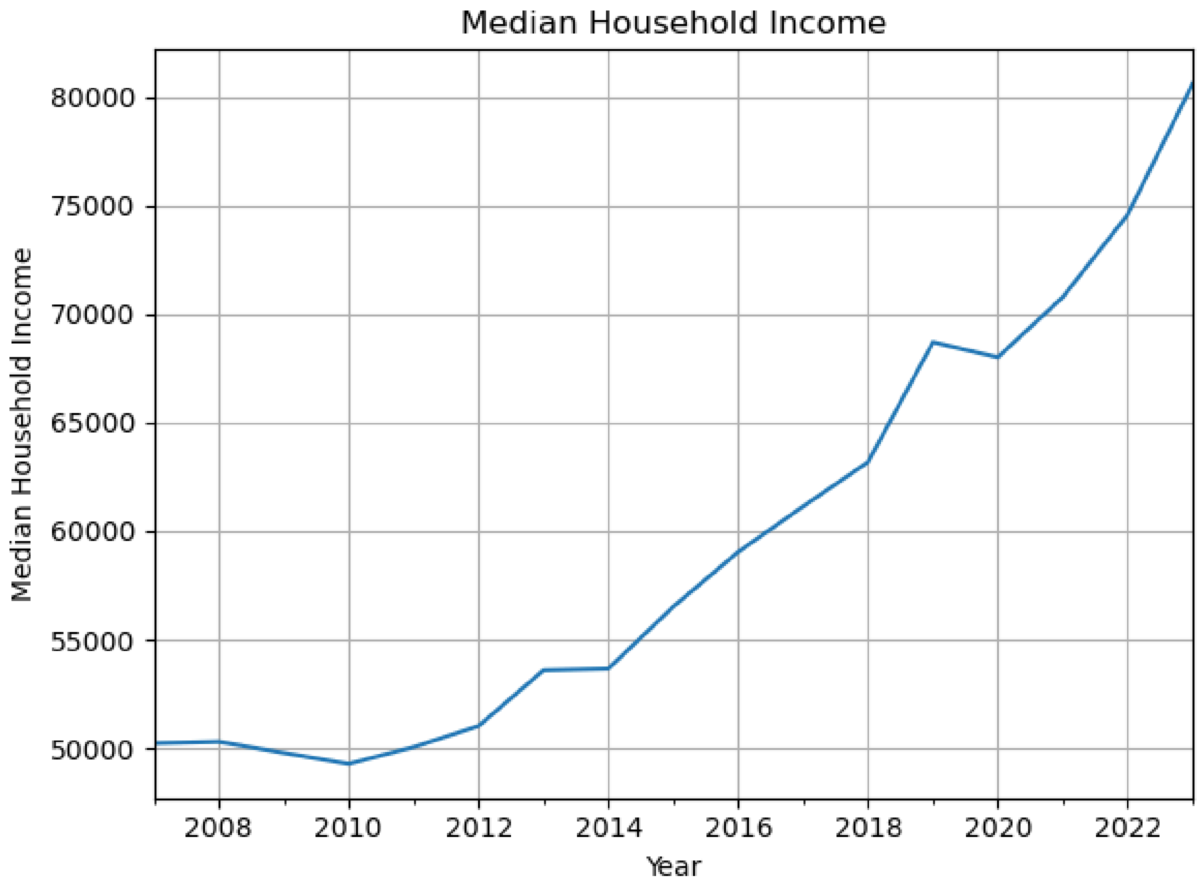
```
# Convert into time series data
df["observation_date"] = pd.to_datetime(df["observation_date"])
df.set_index("observation_date", inplace=True)

# Select numeric columns
df_numeric = df.select_dtypes(include=[np.number])
```

```python
# Time series for each variable
for col in df_numeric.columns:
    plt.figure()
    df_numeric[col].plot(title=col)
    plt.xlabel("Year")
    plt.ylabel(col)
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```

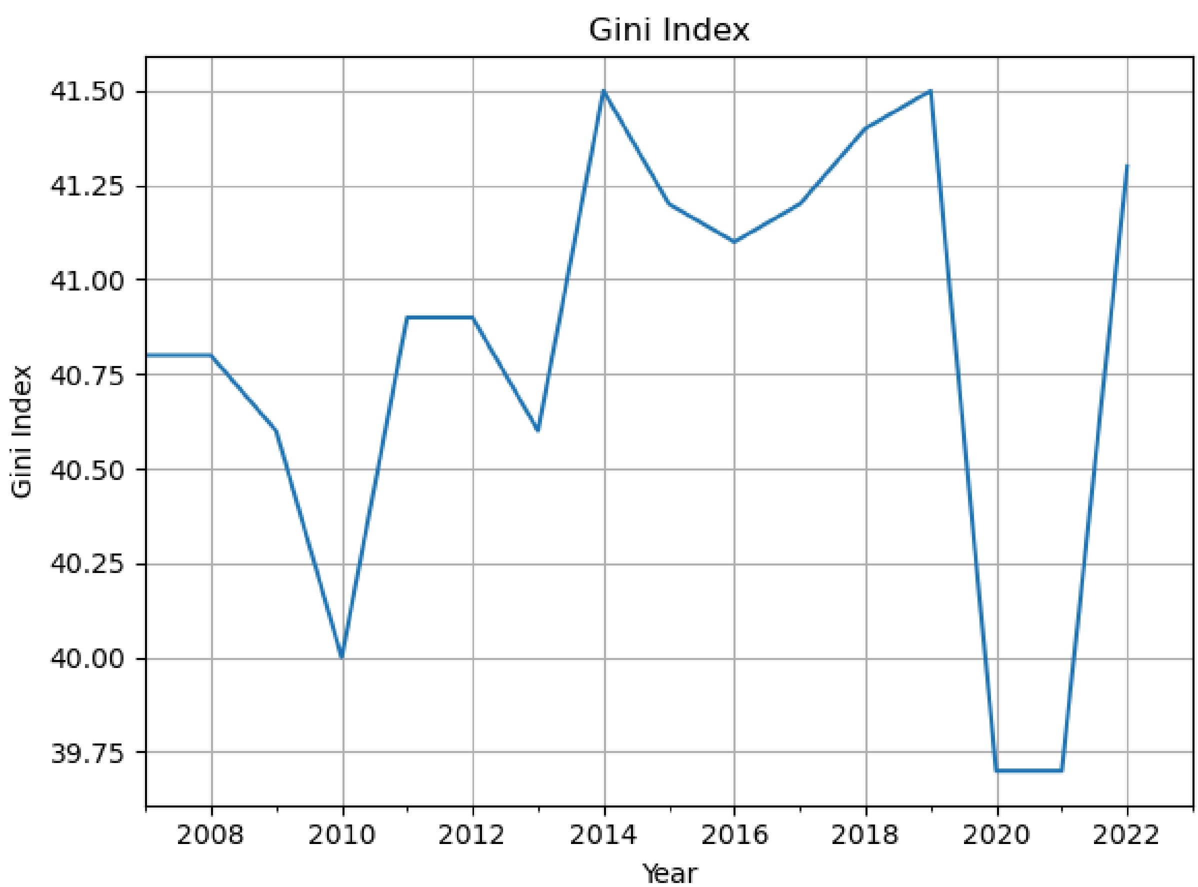



CPI
CPI
Year
1
2
3
4
5
2008
2010
2012
2014
2016
2018
2020
2022

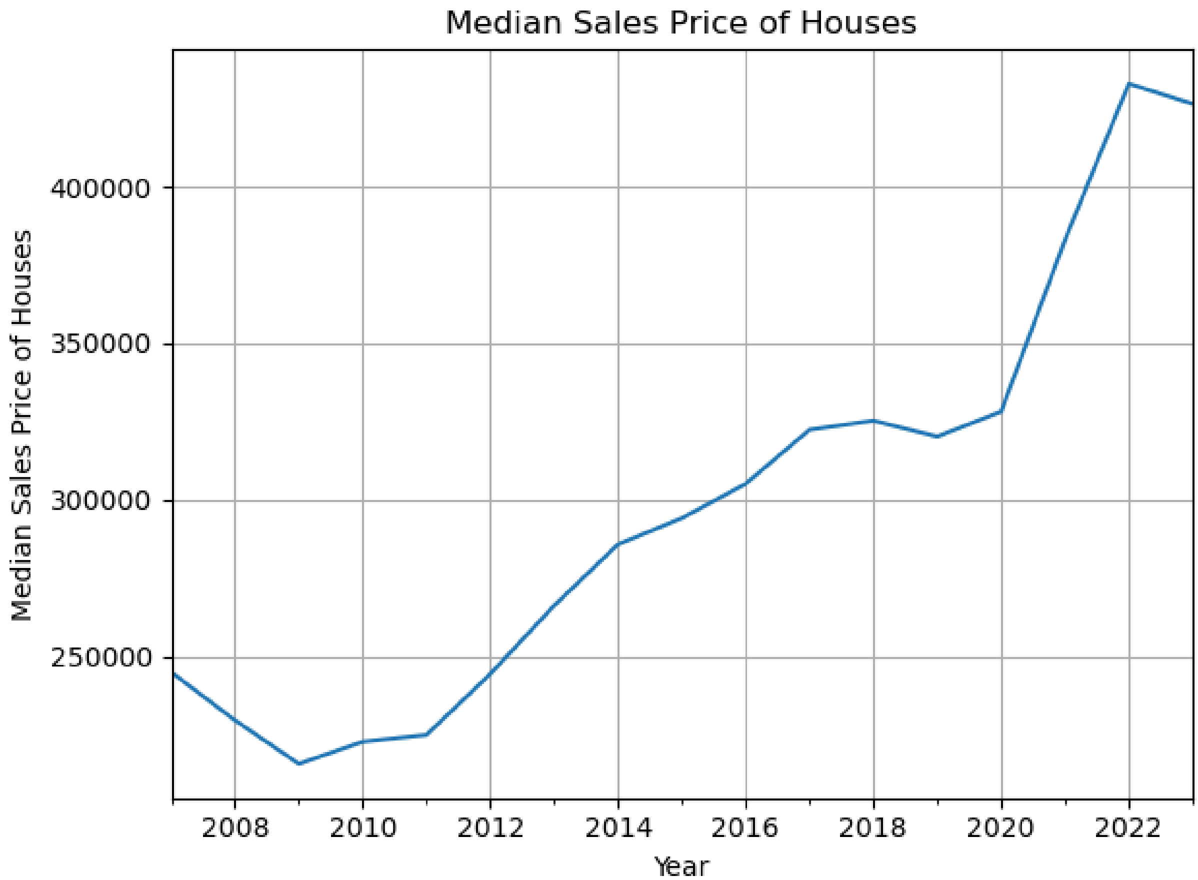

Poverty Rate
Poverty Rate
Year
12.0
12.5
13.0
13.5
14.0
14.5
15.0
15.5
16.0
2008
2010
2012
2014
2016
2018
2020
2022

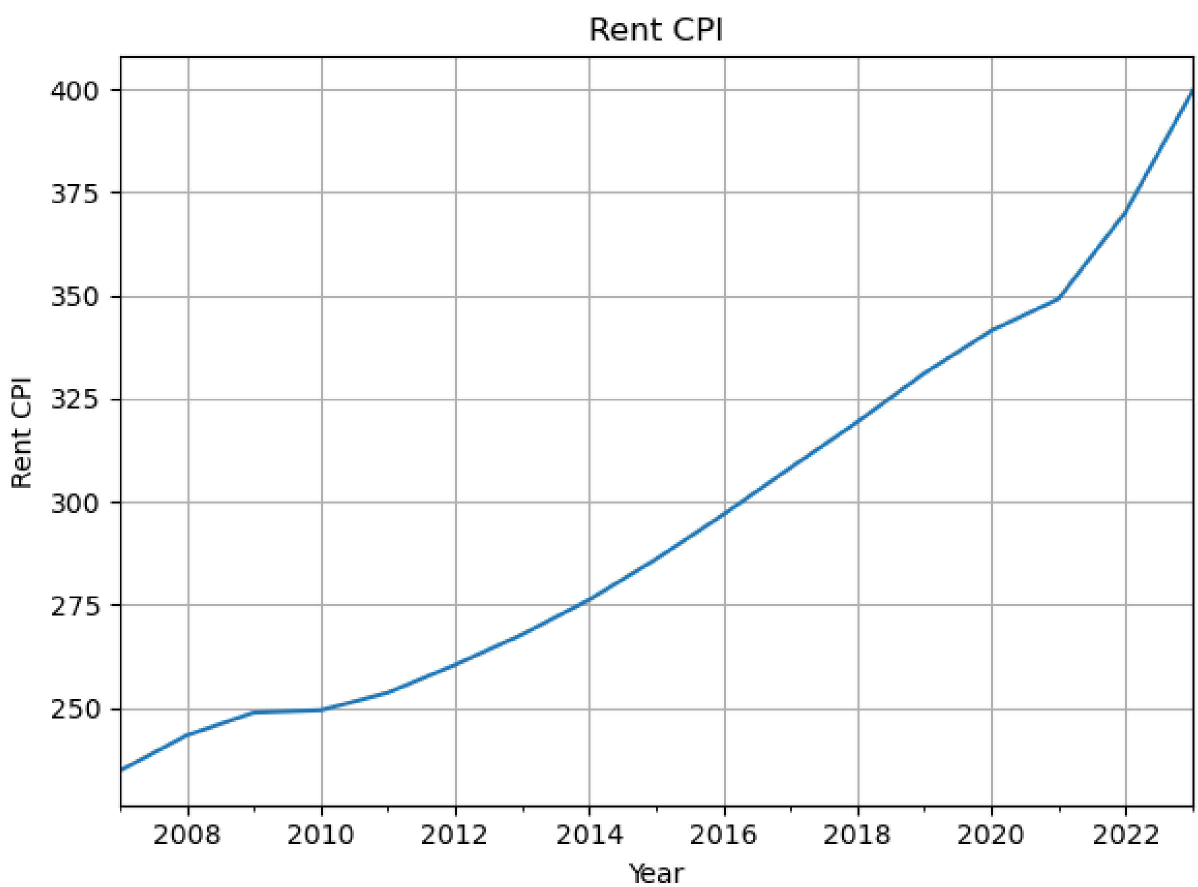


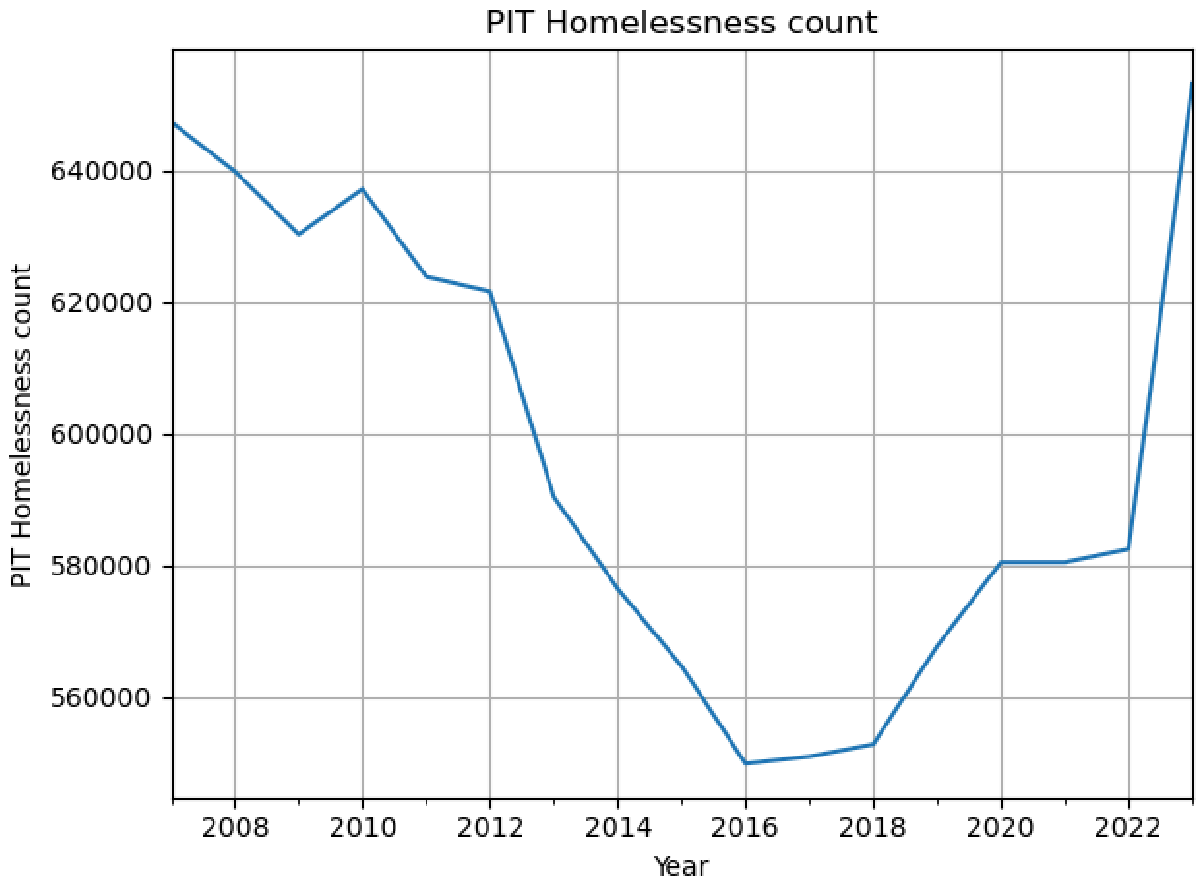
Median Household Income
Median Household Income
80000
75000
70000
65000
60000
55000
50000
2008
2010
2012
2014
2016
2018
2020
2022
Year

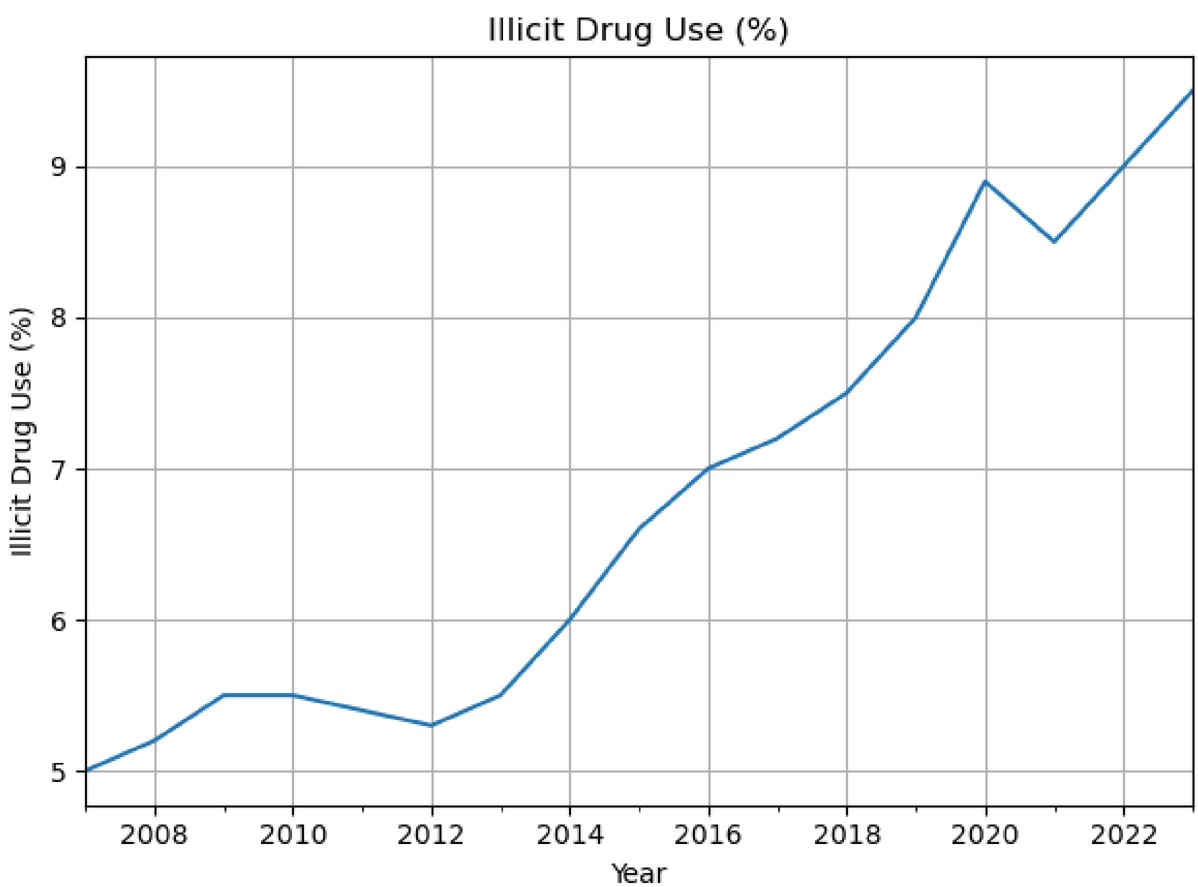


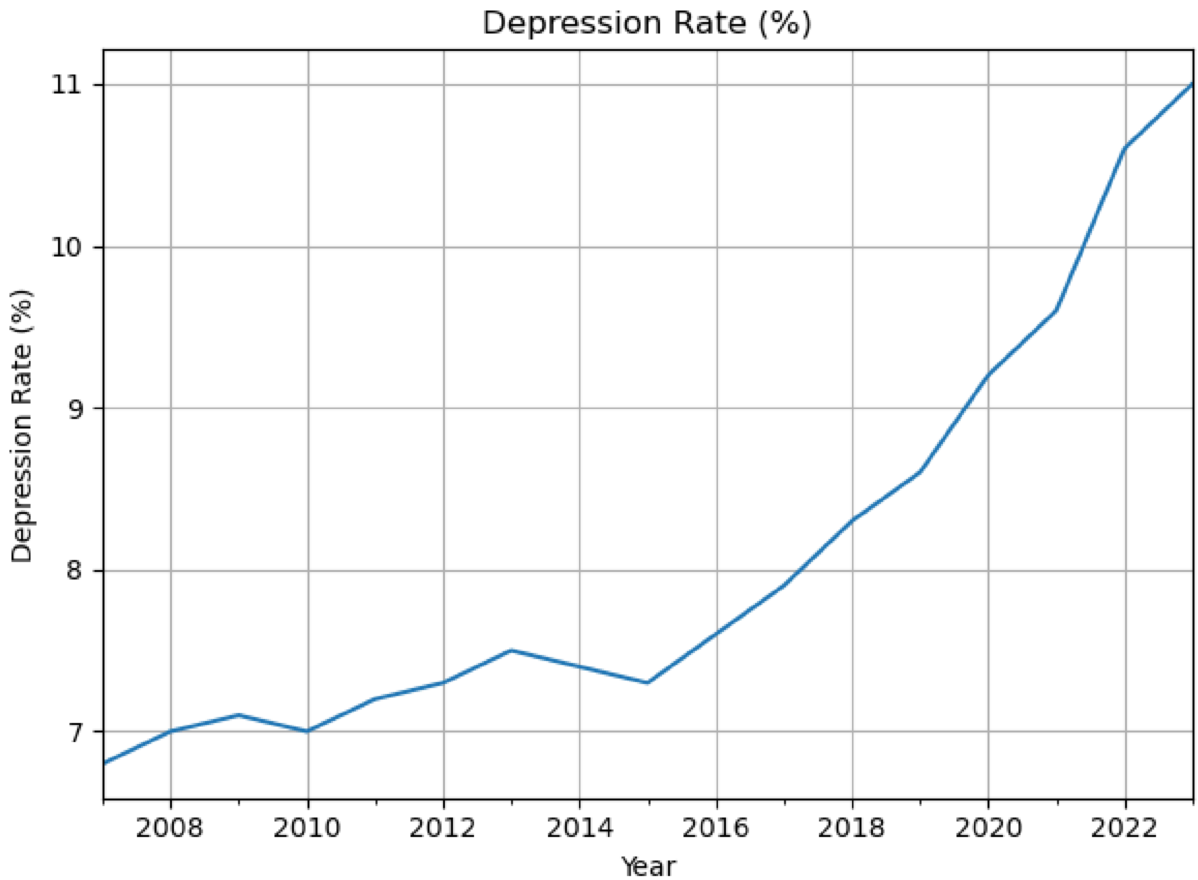
Gini Index
Gini Index
41.50
41.25
41.00
40.75
40.50
40.25
40.00
39.75
2008
2010
2012
2014
2016
2018
2020
2022
Year

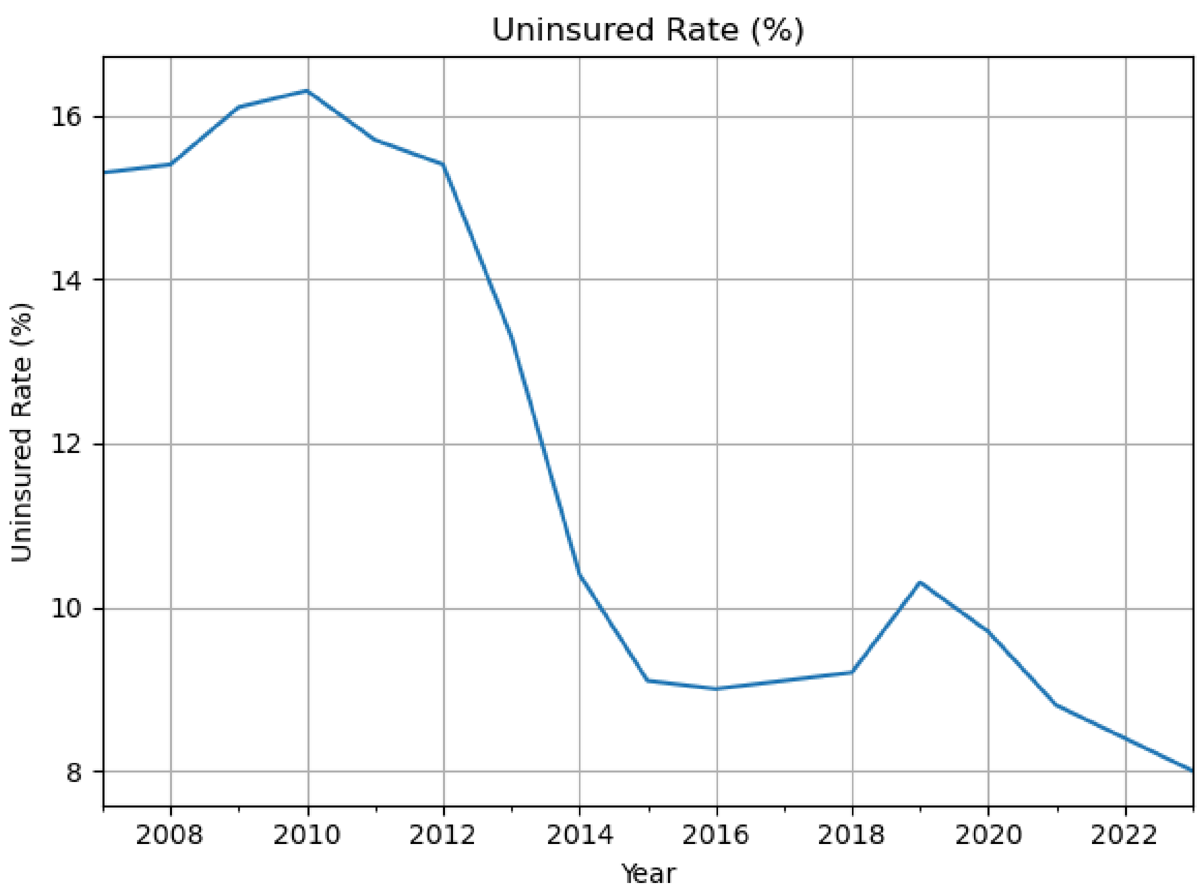

Median Sales Price of Houses
400000
350000
300000
250000
Median Sales Price of Houses
2008
2010
2012
2014
2016
2018
2020
2022
Year


Rent CPI
400
375
350
325
300
275
250
Rent CPI
2008
2010
2012
2014
2016
2018
2020
2022
Year


PIT Homelessness count
PIT Homelessness count
640000
620000
600000
580000
560000
2008
2010
2012
2014
2016
2018
2020
2022
Year

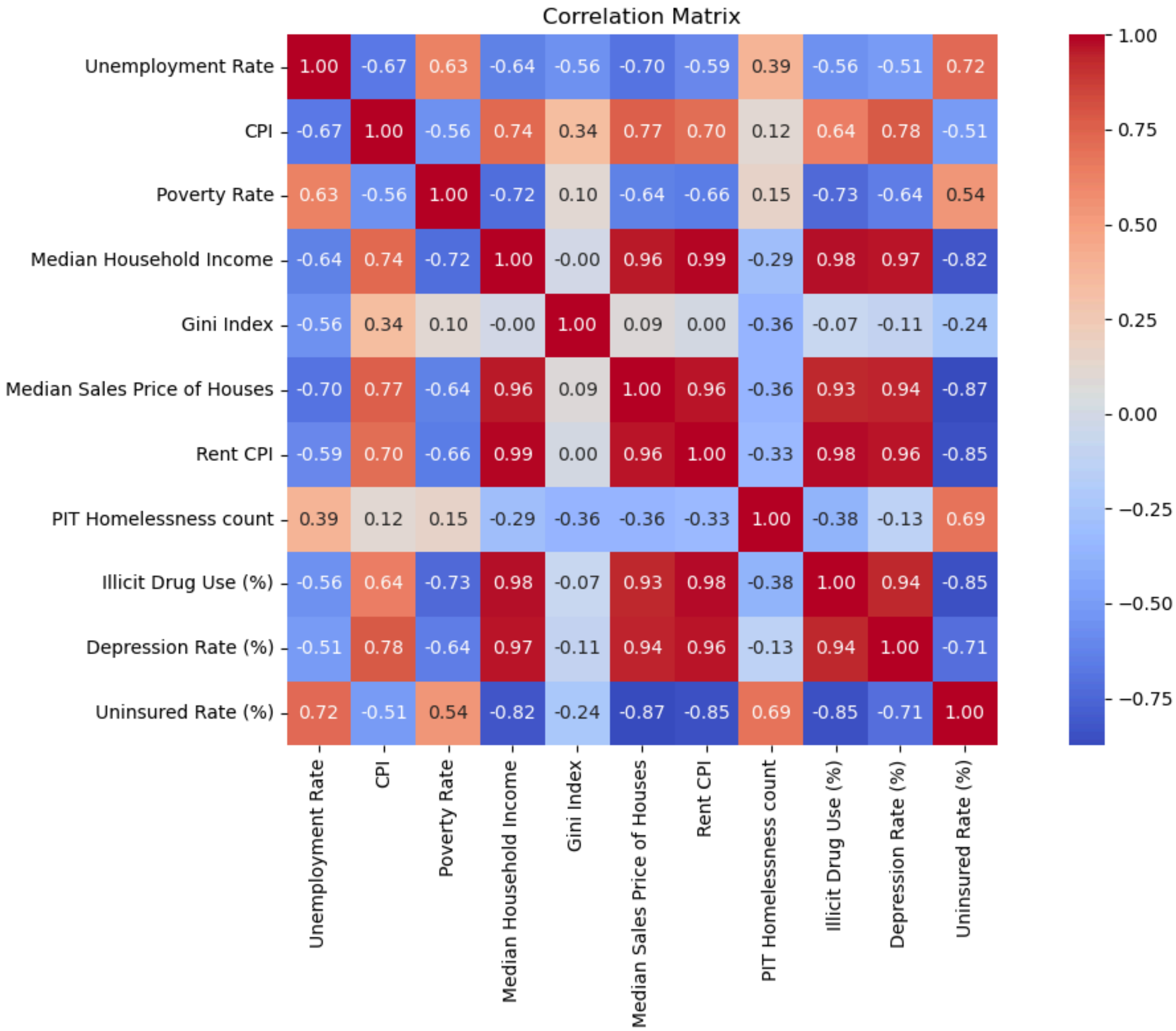

Illicit Drug Use (%)
Illicit Drug Use (%)
9
8
7
6
5
2008
2010
2012
2014
2016
2018
2020
2022
Year


Depression Rate (%)
11
10
9
8
7
Depression Rate (%)
2008
2010
2012
2014
2016
2018
2020
2022
Year


Uninsured Rate (%)
16
14
12
10
8
Uninsured Rate (%)
2008
2010
2012
2014
2016
2018
2020
2022
Year

In [116…

```
# Correlation matrix
correlation_matrix = df_numeric.corr()
plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap="coolwarm", square=True
plt.title("Correlation Matrix")
plt.tight_layout()
plt.show()
```



OLS Regression

We utilized an Ordinary Least Squares (OLS) regression as our first approach to define the baseline linear relationship. The analysis step functioned as a base for assessing how much traditional economic and health indicators explained variations. The preliminary OLS model used three predictors which were Unemployment Rate together with Uninsured Rate (%) and Poverty Rate. The regression model showed an R-squared value of 0.547 which explained that 55% of the variance in homelessness levels resulted from these variables. The Uninsured Rate achieved statistical significance with a p-value of 0.006 whereas the Unemployment Rate and Poverty Rate did not reach statistical significance. The data reveals that access to health services plays a more important role than employment status or income levels in understanding homelessness Statistical implication: Traditional economic variables fail to show significance which could reflect omitted variable bias or the influence of these variables through mechanisms like health..

In [123…

```
# Define dependent and independent variables
y = df_numeric["PIT Homelessness count"]
X = df_numeric[["Unemployment Rate", "Uninsured Rate (%)", "Poverty Rate"]]

# Add Intercept
X = sm.add_constant(X)

# FitOLS model
model = sm.OLS(y, X).fit()

# Print summary
print(model.summary())
```

```
                                OLS Regression Results
==================================================================================
Dep. Variable:     PIT Homelessness count   R-squared:                       0.547
Model:                                OLS   Adj. R-squared:                  0.443
Method:                     Least Squares   F-statistic:                     5.235
Date:                    Sun, 04 May 2025   Prob (F-statistic):             0.0137
Time:                            13:28:35   Log-Likelihood:                -195.31
No. Observations:                      17   AIC:                             398.6
Df Residuals:                          13   BIC:                             402.0
Df Model:                               3
Covariance Type:                nonrobust
==================================================================================
==
                         coef    std err          t      P>|t|      [0.025      0.97
5]
----------------------------------------------------------------------------------
--
const               5.926e+05   7.45e+04      7.954      0.000    4.32e+05    7.54e+
05
Unemployment Rate  -1320.6116   5064.348     -0.261      0.798   -1.23e+04    9620.2
47
Uninsured Rate (%)  1.008e+04   3053.118      3.303      0.006    3487.891    1.67e+
04
Poverty Rate       -7622.5225   6358.678     -1.199      0.252   -2.14e+04    6114.5
67
==============================================================================
Omnibus:                       22.146   Durbin-Watson:                   0.741
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               29.123
Skew:                           1.957   Prob(JB):                     4.74e-07
Kurtosis:                       8.079   Cond. No.                         222.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
```

Final OLS Regression with Log transformation

We log-transformed the homelessness count to resolve constant variance violations before re-estimating the OLS model to enhance its fit. Through this step we deepen our insight into how percentage variations in predictors affect homelessness levels.

The model fit improved and variance became stable when we transformed the dependent variable using its natural logarithm (R-squared = 0.749). Analysis shows that when the Uninsured Rate and Depression Rate each rise by 1%, homelessness experiences an estimated 2–3% increase in its logged valu.

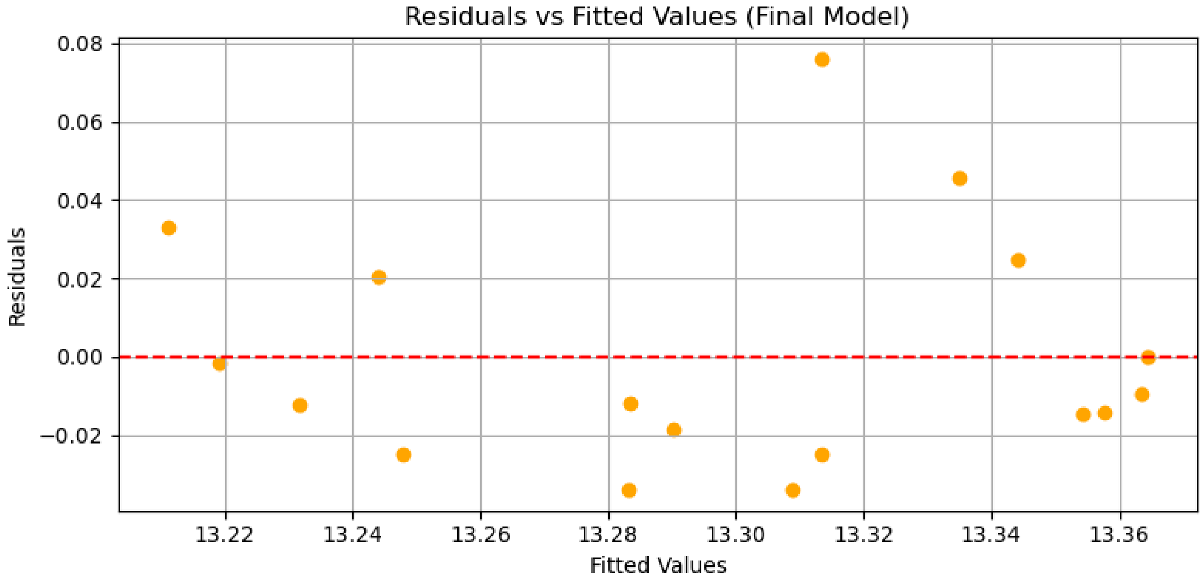
Research implication: The analysis demonstrates that health-related predictors are significant factors in this model. Log transformation proves vital for decoding marginal percentage changes and adhering to linear model assumptions.

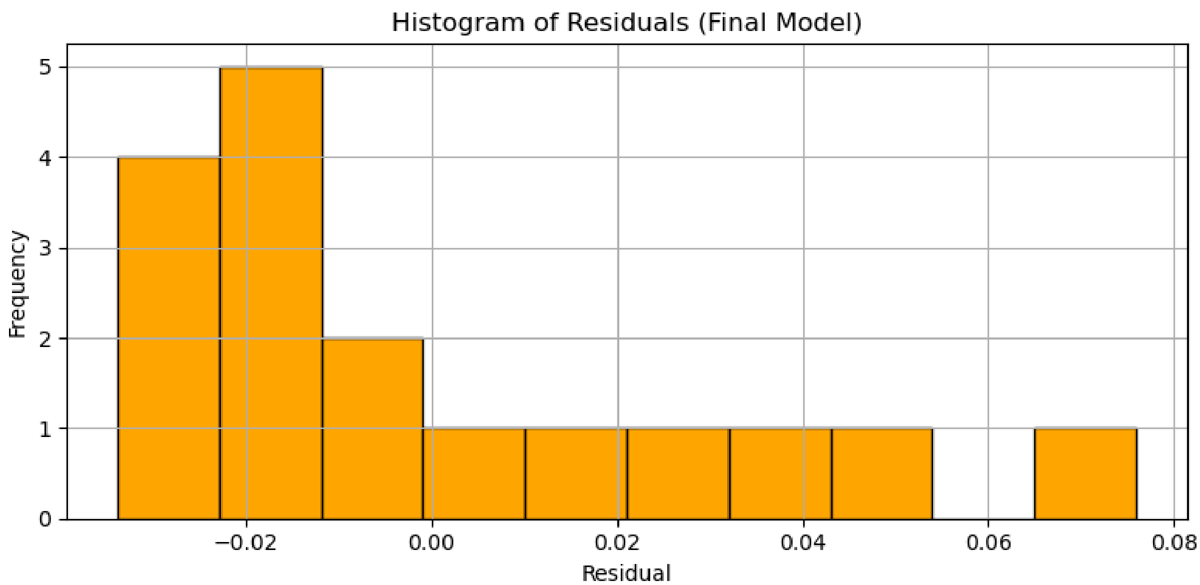
In [125…

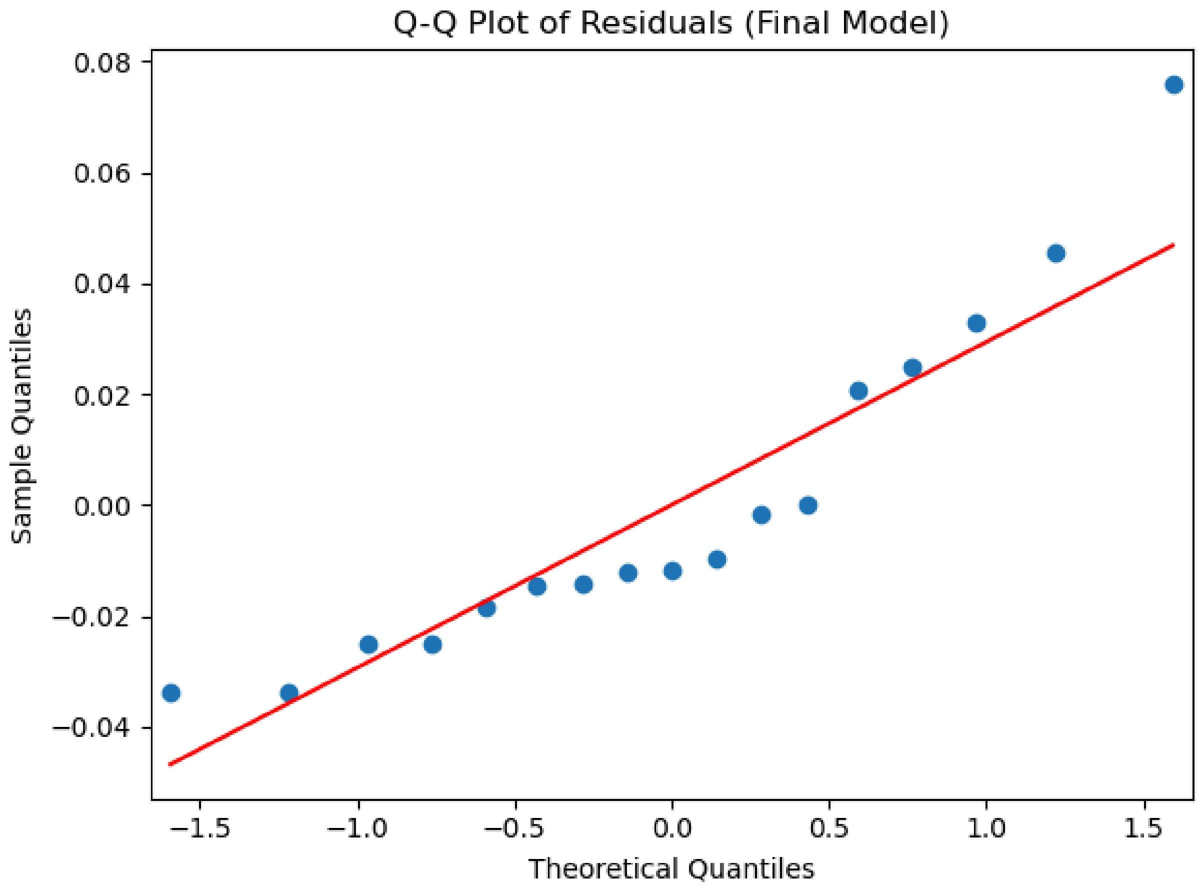
```python
# Recreate the X and y
df = pd.read_csv(r"C:\Users\matia\Downloads\homelessness_data.csv")
df["observation_date"] = pd.to_datetime(df["observation_date"])
df.set_index("observation_date", inplace=True)
df_numeric = df.select_dtypes(include=[np.number])
y_log = np.log(df_numeric["PIT Homelessness count"])

# Define predictors
X_final = df_numeric[["Uninsured Rate (%)", "Depression Rate (%)"]]
X_final = sm.add_constant(X_final)

# Fit the final OLS model
model_final = sm.OLS(y_log, X_final).fit()
print(model_final.summary())

# Residual diagnostics
residuals = model_final.resid
fitted = model_final.fittedvalues

# 1. Residuals vs Fitted
plt.figure(figsize=(8, 4))
plt.scatter(fitted, residuals, color='orange')
plt.axhline(0, color='red', linestyle='--')
plt.title("Residuals vs Fitted Values (Final Model)")
plt.xlabel("Fitted Values")
plt.ylabel("Residuals")
plt.grid(True)
plt.tight_layout()
plt.show()

# 2. Histogram of residuals
plt.figure(figsize=(8, 4))
plt.hist(residuals, bins=10, edgecolor='black', color='orange')
plt.title("Histogram of Residuals (Final Model)")
```

```python
plt.xlabel("Residual")
plt.ylabel("Frequency")
plt.grid(True)
plt.tight_layout()
plt.show()

# 3. Q-Q Plot
qqplot(residuals, line='s')
plt.title("Q-Q Plot of Residuals (Final Model)")
plt.tight_layout()
plt.show()
```

```
                            OLS Regression Results
==================================================================================
Dep. Variable:     PIT Homelessness count   R-squared:                       0.749
Model:                                OLS   Adj. R-squared:                  0.713
Method:                     Least Squares   F-statistic:                     20.84
Date:                    Sun, 04 May 2025   Prob (F-statistic):           6.35e-05
Time:                            13:29:06   Log-Likelihood:                 35.855
No. Observations:                      17   AIC:                            -65.71
Df Residuals:                          14   BIC:                            -63.21
Df Model:                               2
Covariance Type:                nonrobust
====================================================================================
===
                          coef    std err          t      P>|t|      [0.025      0.9
75]
------------------------------------------------------------------------------------
---
const                  12.7524      0.106    120.056      0.000      12.525      12.
980
Uninsured Rate (%)      0.0228      0.004      6.384      0.000       0.015       0.
030
Depression Rate (%)     0.0345      0.009      3.869      0.002       0.015       0.
054
==============================================================================
Omnibus:                        5.635   Durbin-Watson:                   1.270
Prob(Omnibus):                  0.060   Jarque-Bera (JB):                3.463
Skew:                           1.087   Prob(JB):                        0.177
Kurtosis:                       3.403   Cond. No.                         196.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
```


Residuals vs Fitted Values (Final Model)
Residuals
Fitted Values
0.08
0.06
0.04
0.02
0.00
−0.02
13.22
13.24
13.26
13.28
13.30
13.32
13.34
13.36


Histogram of Residuals (Final Model)
Frequency
Residual
5
4
3
2
1
0
−0.02
0.00
0.02
0.04
0.06
0.08



Lasso Regression

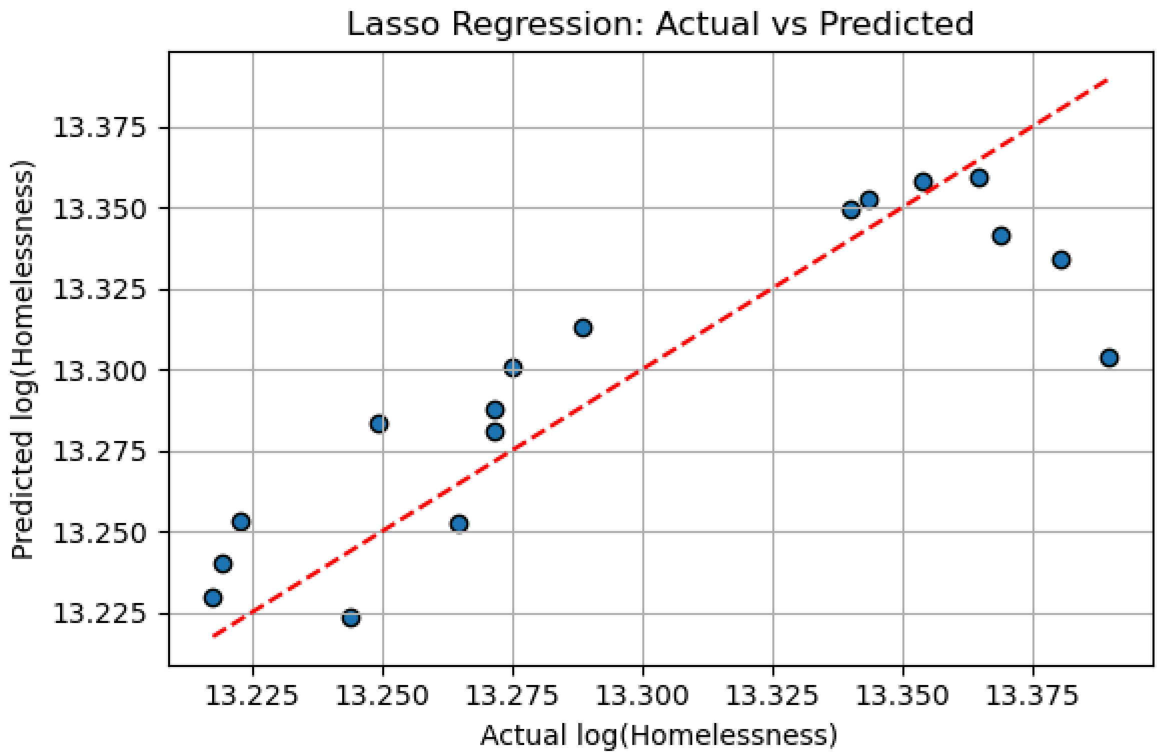
To avoid overfitting and select the most predictive features from a larger set, we employed Lasso regression next. This regularization technique is a predictor shrinkage as well as variable selection method. The most 2 relevant predictors were Uninsured Rate and Depression Rate (Lasso regression). The shrinkage worked by zeroing out coefficients for unassociated variables that improved interpretability and reduced overfitting. Intuition: Lasso does variable selection and regularization and is particularly useful in high-p, low-n settings. Statistical consequence: Lasso does variable selection and regularization, especially useful when n << p. From a research standpoint, it can help target interventions by flagging the risk factors that matter most.

In [135…

```
# Reload and prepare the data
df = pd.read_csv(r"C:\Users\matia\Downloads\homelessness_data.csv")
df["observation_date"] = pd.to_datetime(df["observation_date"])
df.set_index("observation_date", inplace=True)
df_numeric = df.select_dtypes(include=[np.number])
y_log = np.log(df_numeric["PIT Homelessness count"])

# Select predictors
X = df_numeric[["Uninsured Rate (%)", "Depression Rate (%)"]]

# Create a pipeline that scales the data then applies LassoCV
```

```python
pipeline = make_pipeline(
    StandardScaler(),
    LassoCV(cv=5, random_state=0)
)

# Fit pipeline
pipeline.fit(X, y_log)

# Extract model and results
lasso_model = pipeline.named_steps['lassocv']
y_pred = pipeline.predict(X)
r2 = r2_score(y_log, y_pred)
mse = mean_squared_error(y_log, y_pred)

# Print results
print("Optimal alpha (λ):", lasso_model.alpha_)
print("R-squared:", r2)
print("Mean Squared Error:", mse)
print("\nLasso Coefficients:")
for feature, coef in zip(X.columns, lasso_model.coef_):
    print(f"{feature}: {coef:.5f}")

# Plot actual vs predicted values
plt.figure(figsize=(6, 4))
plt.scatter(y_log, y_pred, edgecolor='k')
plt.plot([y_log.min(), y_log.max()], [y_log.min(), y_log.max()], 'r--')
plt.xlabel("Actual log(Homelessness)")
plt.ylabel("Predicted log(Homelessness)")
plt.title("Lasso Regression: Actual vs Predicted")
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
Optimal alpha (λ): 0.0024886344717958927
R-squared: 0.7360191336932349
Mean Squared Error: 0.0009051712759999537

Lasso Coefficients:
Uninsured Rate (%): 0.06269
Depression Rate (%): 0.03459
```



Feuture Selection

To widen the Lasso model, we added all predictors available to investigate which variables were frequently kept. This is a step towards a better comprehension of which dimensions of the socioeconomic space contain more predictive power. Applying LassoCV to all predictors suggested that Uninsured Rate, Gini Index, and CPI persisted. The sparsity of this model is desirable for model parsimony and indicates that inequality and inflation, in addition to health access, are correlates of homelessness trends. Statistical interpretation: The variable selection helps prevent overfitting. Research implications: homelessness models should not disregard inequality indices.

In [137…

```
# Reload your dataset
df = pd.read_csv(r"C:\Users\matia\Downloads\homelessness_data.csv")
df["observation_date"] = pd.to_datetime(df["observation_date"])
df.set_index("observation_date", inplace=True)

# Define predictors and target
y = np.log(df["PIT Homelessness count"])
X = df.select_dtypes(include=[np.number]).drop("PIT Homelessness count", axis=1)

# Pipeline
pipeline_lasso = Pipeline([
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler()),
    ('lasso', LassoCV(cv=5, random_state=0, max_iter=10000))
])
```

```
# Fit pipeline
pipeline_lasso.fit(X, y)

# Output
print("Lasso Selected Features and Coefficients:")
for feature, coef in zip(X.columns, pipeline_lasso.named_steps['lasso'].coef_):
    print(f"{feature}: {coef:.5f}")
```

```
Lasso Selected Features and Coefficients:
Unemployment Rate: 0.00000
CPI: 0.02885
Poverty Rate: 0.00000
Median Household Income: 0.00000
Gini Index: -0.01110
Median Sales Price of Houses: -0.00000
Rent CPI: 0.00000
Illicit Drug Use (%): -0.00000
Depression Rate (%): 0.00000
Uninsured Rate (%): 0.04776
```

Double Machine Learning

Once we had identified the main predictors from the previous two stages with both OLS and Lasso, we used Double Machine Learning (DML) to estimate causal effects flexibly adjusting for confounders. This is a transition from associatived to causitive thinking. Leveraging the `econml` package, we applied a semiparametric DML framework to estimate the ATE of Uninsured Rate on homelessness, controlling for Depression Rate. The ATE was 0.0147 (95% CI = 0.0115–0.0180), which demonstrated causality. Role reversal indicated a lack of or an only weak effect of Depression Rate.

Statistical insight: cross-fitting is fun, and DML controls confounding using flexible learners. The approach synthesized causal inference under the influence of observational data. Implication for research: health insurance can be considered causally bordering for the decrease in homelessness.

In [139…

```
# Reload and prepare the data
df = pd.read_csv(r"C:\Users\matia\Downloads\homelessness_data.csv")
df["observation_date"] = pd.to_datetime(df["observation_date"])
df.set_index("observation_date", inplace=True)
df_numeric = df.select_dtypes(include=[np.number])

# Outcome and treatment
Y = np.log(df_numeric["PIT Homelessness count"])  # log outcome
T = df_numeric["Uninsured Rate (%)"]  # treatment variable

# Controls
X = df_numeric[["Depression Rate (%)"]]  # confounders

# Define models
model_y = RandomForestRegressor(n_estimators=100, random_state=0)
model_t = LassoCV(cv=5, random_state=0)
```

```
# Set up and fit the DML model
dml = LinearDML(model_y=clone(model_y),
                model_t=clone(model_t),
                discrete_treatment=False,
                cv=KFold(n_splits=5, shuffle=True, random_state=0),
                random_state=0)

dml.fit(Y, T, X=X)

# Estimate average treatment effect (ATE) and 95% confidence interval
effects = dml.const_marginal_effect(X).flatten()
ate = np.mean(effects)
se = np.std(effects, ddof=1) / np.sqrt(len(effects))
z = stats.norm.ppf(0.975)
ci_lower = ate - z * se
ci_upper = ate + z * se

# Results
print("Double Machine Learning Results (Unconfoundedness):")
print("ATE (mean marginal effect of Uninsured Rate on log Homelessness):", ate)
print("95% Confidence Interval:", (ci_lower, ci_upper))
```

```
Double Machine Learning Results (Unconfoundedness):
ATE (mean marginal effect of Uninsured Rate on log Homelessness): 0.0147146465242126
35
95% Confidence Interval: (0.011457992885708488, 0.01797130016271678)
```

In [141…

```
# Reload and prepare the data
df = pd.read_csv(r"C:\Users\matia\Downloads\homelessness_data.csv")
df["observation_date"] = pd.to_datetime(df["observation_date"])
df.set_index("observation_date", inplace=True)
df_numeric = df.select_dtypes(include=[np.number])

# Outcome
Y = np.log(df_numeric["PIT Homelessness count"])

# New Treatment and Control
T = df_numeric["Depression Rate (%)"]
X = df_numeric[["Uninsured Rate (%)"]]

# Define models
model_y = RandomForestRegressor(n_estimators=100, random_state=0)
model_t = LassoCV(cv=5, random_state=0)

# it model
dml = LinearDML(model_y=clone(model_y),
                model_t=clone(model_t),
                discrete_treatment=False,
                cv=KFold(n_splits=5, shuffle=True, random_state=0),
                random_state=0)

dml.fit(Y, T, X=X)

# Estimate ATE and 95% CI
effects = dml.const_marginal_effect(X).flatten()
```

```
ate = np.mean(effects)
se = np.std(effects, ddof=1) / np.sqrt(len(effects))
z = stats.norm.ppf(0.975)
ci_lower = ate - z * se
ci_upper = ate + z * se

# Results
print("Double Machine Learning Results (Depression Rate → log Homelessness):")
print("ATE (mean marginal effect of Depression Rate):", ate)
print("95% Confidence Interval:", (ci_lower, ci_upper))
```

```
Double Machine Learning Results (Depression Rate → log Homelessness):
ATE (mean marginal effect of Depression Rate): -0.008449677723821672
95% Confidence Interval: (-0.018699121320488895, 0.001799765872845551)
```

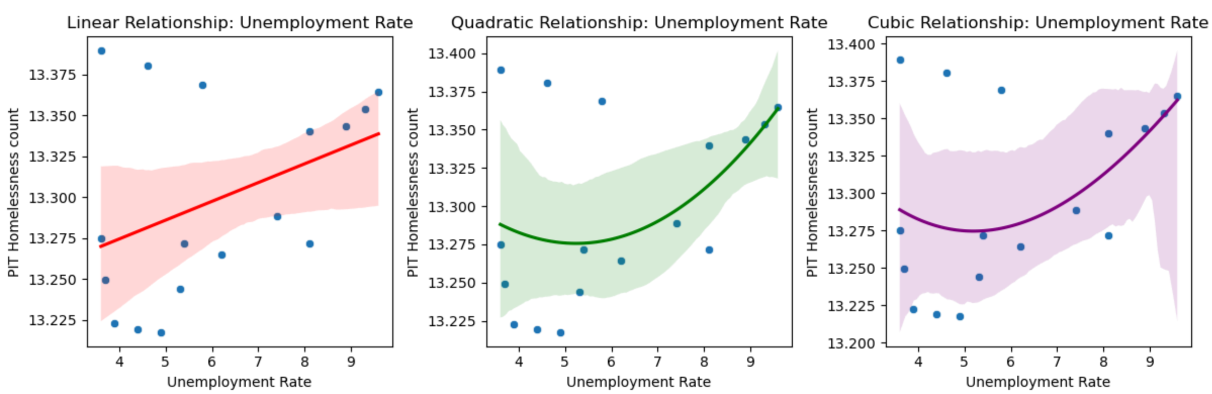
Polynomial Relationships

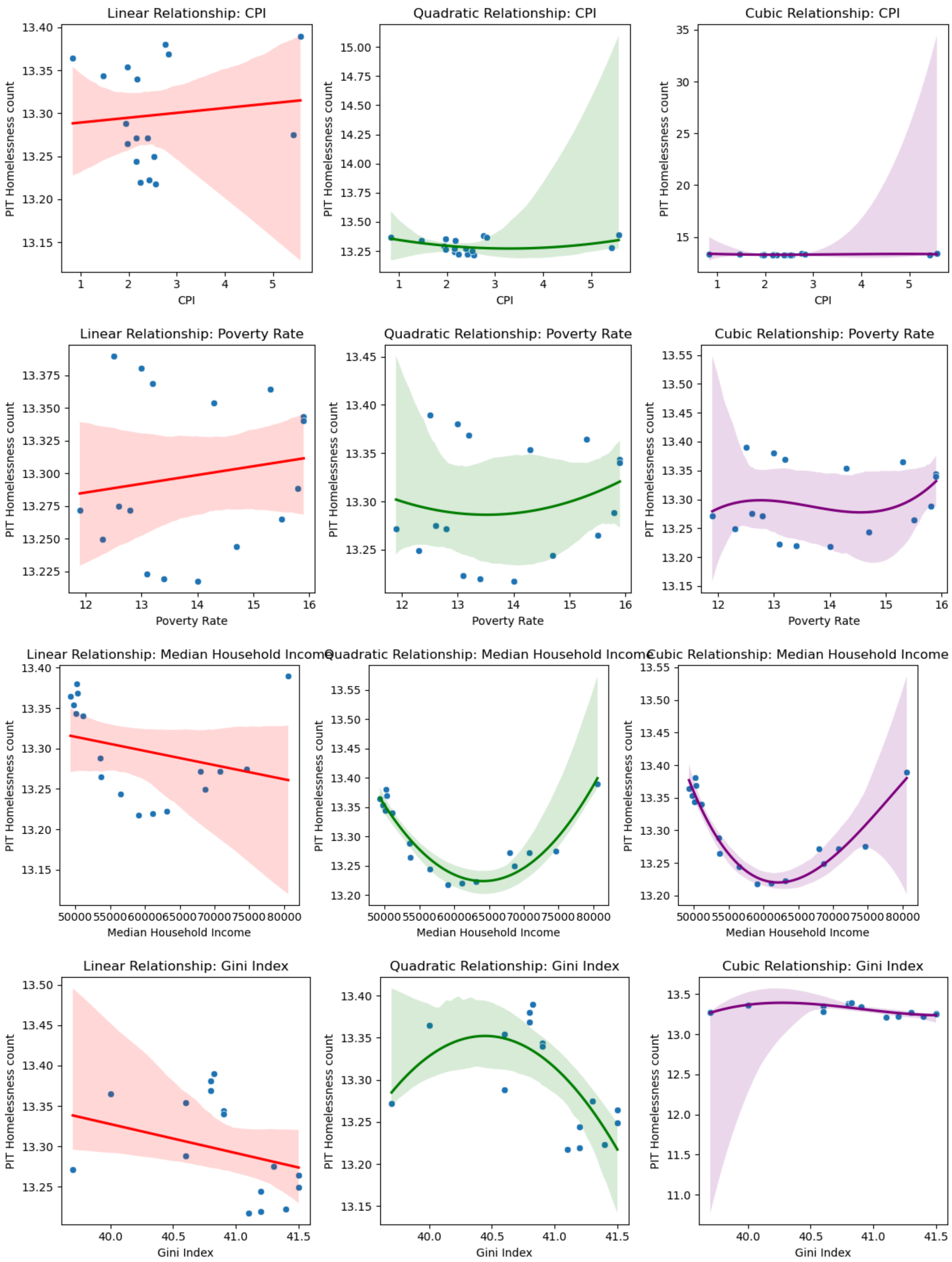
Given that the association between the predictors and the outcome may not be purely linear, we then examined polynomial relationships. This procedure simulates possible threshold and saturation effects not captured by linear models. This exploration, a self-terminated search for non-linear relations, showed polynomial fits between predictors and homelessness. For instance, the influence of Uninsured Rate in other direction past a certain point was sharper. This gives support to the notion of tipping points in homelessness dynamics. Statistically, this provides a rationale for extending to beyond linear models. In terms of research, it suggests that policy interventions can have outsize effects once certain thresholds are reached.

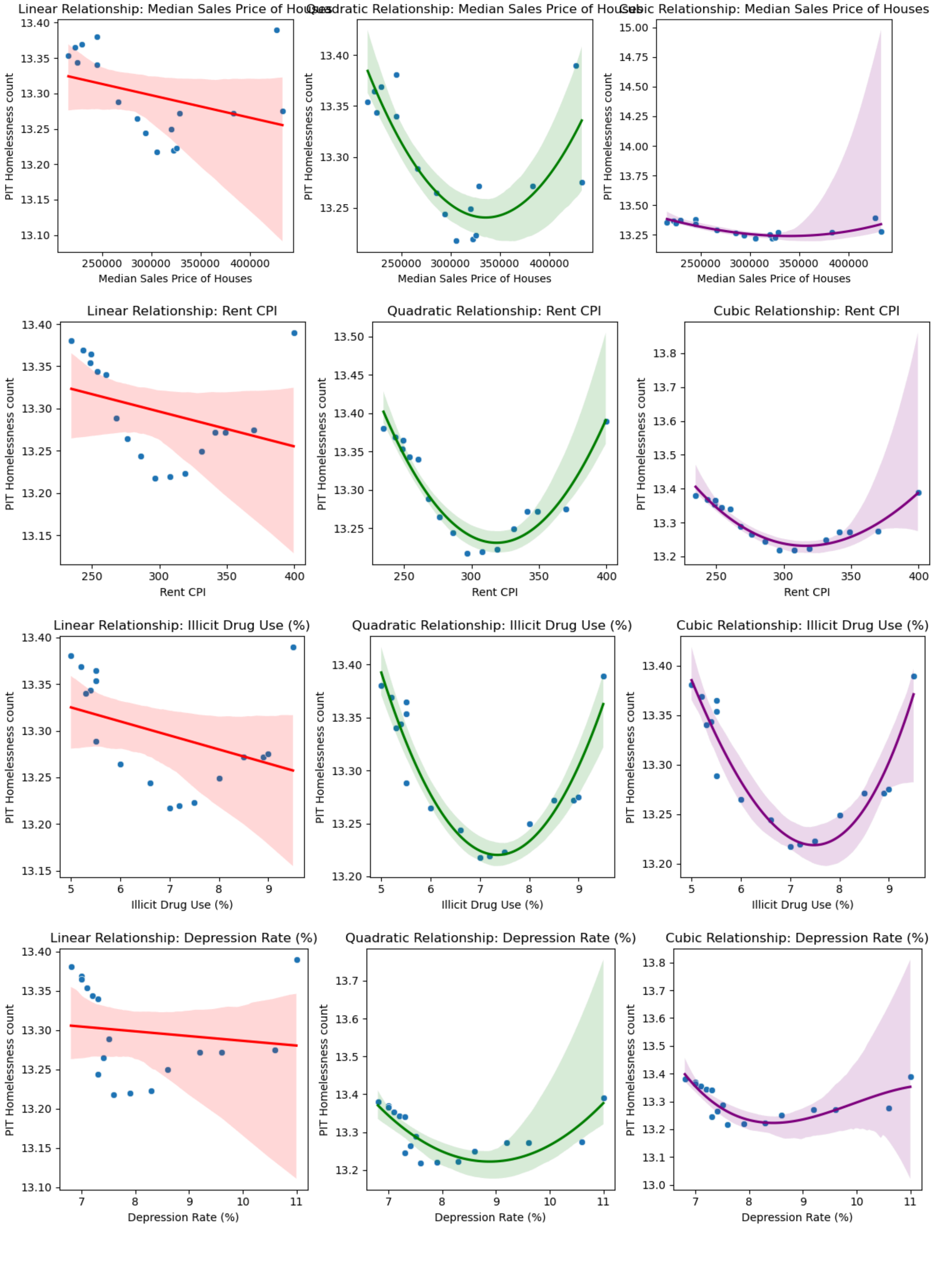
In [143…

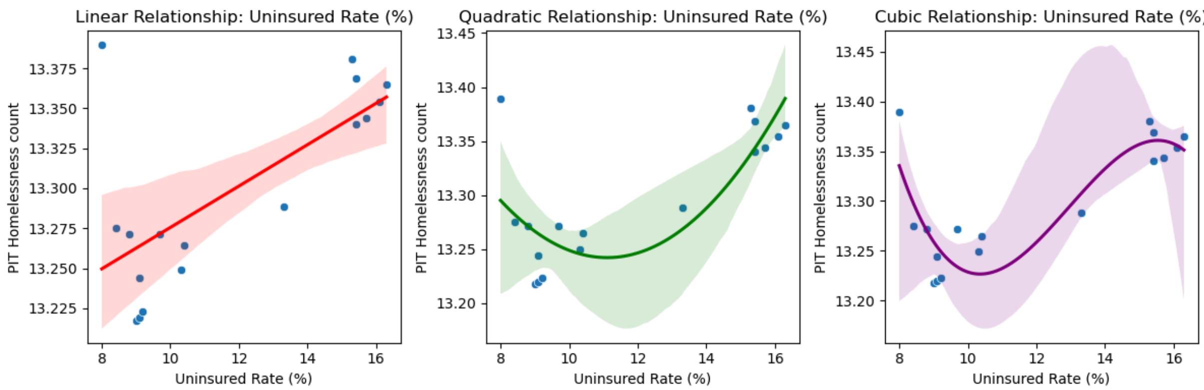
```
df = pd.read_csv(r"C:\Users\matia\Downloads\homelessness_data.csv")
df["observation_date"] = pd.to_datetime(df["observation_date"])
df.set_index("observation_date", inplace=True)

# Target and features
y = np.log(df["PIT Homelessness count"])
X = df.select_dtypes(include=[np.number]).drop("PIT Homelessness count", axis=1)

# Missing values
imputer = SimpleImputer(strategy='mean')
X_imputed = pd.DataFrame(imputer.fit_transform(X), columns=X.columns, index=X.index

# Check polynomial relationships
for feature in X_imputed.columns:
    plt.figure(figsize=(12,4))

    # Linear plot
    plt.subplot(1, 3, 1)
    sns.scatterplot(x=X_imputed[feature], y=y)
    sns.regplot(x=X_imputed[feature], y=y, scatter=False, color='red')
    plt.title(f'Linear Relationship: {feature}')

    # Quadratic plot
    plt.subplot(1, 3, 2)
```

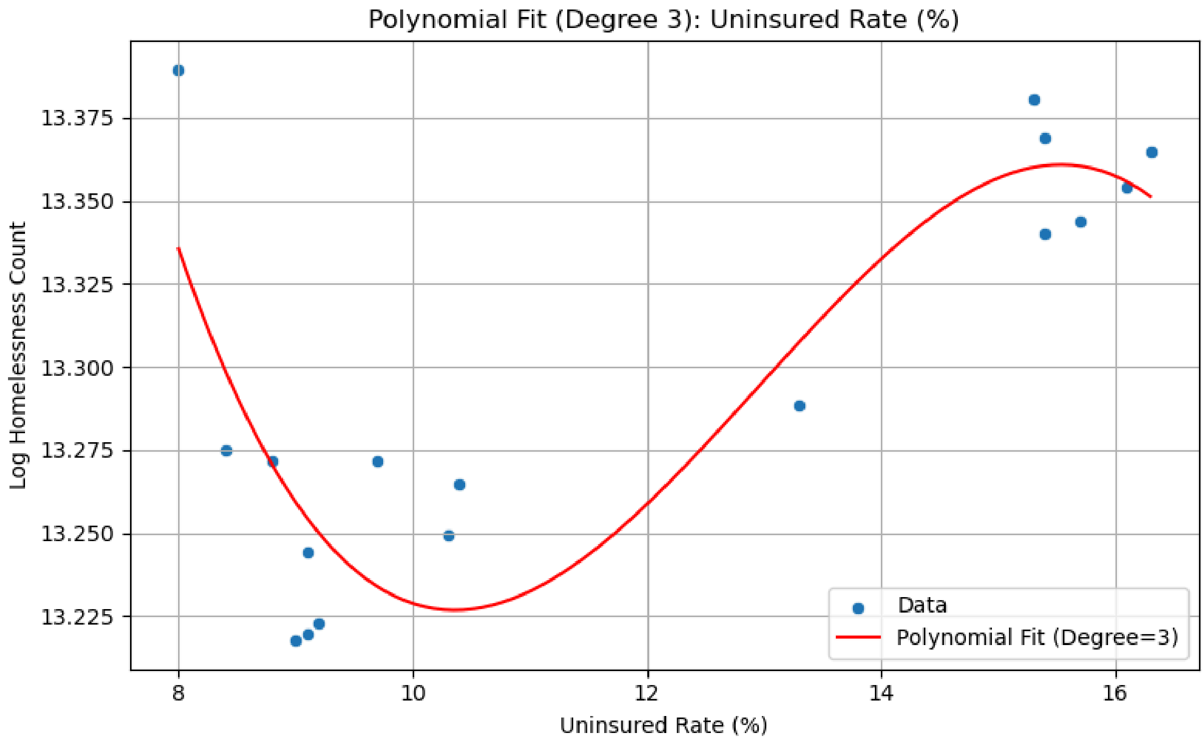
```python
    sns.scatterplot(x=X_imputed[feature], y=y)
    sns.regplot(x=X_imputed[feature], y=y, scatter=False, order=2, color='green')
    plt.title(f'Quadratic Relationship: {feature}')

    # Cubic plot
    plt.subplot(1, 3, 3)
    sns.scatterplot(x=X_imputed[feature], y=y)
    sns.regplot(x=X_imputed[feature], y=y, scatter=False, order=3, color='purple')
    plt.title(f'Cubic Relationship: {feature}')

    plt.tight_layout()
    plt.show()


key_feature = "Uninsured Rate (%)"

pipeline_poly = Pipeline([
    ('poly', PolynomialFeatures(degree=3, include_bias=False)),
    ('scaler', StandardScaler()),
    ('linreg', LinearRegression())
])

X_poly = X_imputed[[key_feature]]
pipeline_poly.fit(X_poly, y)

# Plot polynomial fit
x_vals = np.linspace(X_poly.min(), X_poly.max(), 100)
y_pred = pipeline_poly.predict(x_vals)

plt.figure(figsize=(8, 5))
sns.scatterplot(x=X_poly[key_feature], y=y, label='Data')
plt.plot(x_vals, y_pred, color='red', label='Polynomial Fit (Degree=3)')
plt.xlabel(key_feature)
plt.ylabel('Log Homelessness Count')
plt.title(f'Polynomial Fit (Degree 3): {key_feature}')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```




Linear Relationship: CPI
Quadratic Relationship: CPI
Cubic Relationship: CPI
Linear Relationship: Poverty Rate
Quadratic Relationship: Poverty Rate
Cubic Relationship: Poverty Rate
Linear Relationship: Median Household Income
Quadratic Relationship: Median Household Income
Cubic Relationship: Median Household Income
Linear Relationship: Gini Index
Quadratic Relationship: Gini Index
Cubic Relationship: Gini Index
PIT Homelessness count
CPI
Poverty Rate
Median Household Income
Gini Index


Linear Relationship: Median Sales Price of Houses
Quadratic Relationship: Median Sales Price of Houses
Cubic Relationship: Median Sales Price of Houses
Median Sales Price of Houses
Linear Relationship: Rent CPI
Quadratic Relationship: Rent CPI
Cubic Relationship: Rent CPI
Rent CPI
Linear Relationship: Illicit Drug Use (%)
Quadratic Relationship: Illicit Drug Use (%)
Cubic Relationship: Illicit Drug Use (%)
Illicit Drug Use (%)
Linear Relationship: Depression Rate (%)
Quadratic Relationship: Depression Rate (%)
Cubic Relationship: Depression Rate (%)
Depression Rate (%)
PIT Homelessness count





Non Linear models

A range of non-linear machine learning models were used to improve prediction and facilitate flexible modeling of complex interactions. This complements causal inference by validating on out-of-sample fit and ranking the importance of the variables. We took Random Forest, Gradient Boosting, Support Vector Regression, as well as KNN in comparison. Gradient Boosting demonstrated the smallest mean absolute error (MAE = 0.0164). Depression Rate and Uninsured Rate were the two most important predictors in this model based on their feature importances. Statistical upshot: Ensemble models are a clairvoyant's dream, at least in small data sets. Implications of research: these models help predict future homelessness under different policy interventions.

In [144…

```
# Prepare the data
X = df.select_dtypes(include=[np.number]).drop("PIT Homelessness count", axis=1)
y = np.log(df["PIT Homelessness count"])
```

```
X = SimpleImputer(strategy='mean').fit_transform(X)
X = StandardScaler().fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Model
rf = RandomForestRegressor(n_estimators=100, random_state=0)
rf.fit(X_train, y_train)

# Predict
y_pred = rf.predict(X_test)
print("Random Forest MAE:", mean_absolute_error(y_test, y_pred))
```

```
Random Forest MAE: 0.03084642045001562
```

In [145...

```
gbr = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1, random_state=0
gbr.fit(X_train, y_train)

y_pred = gbr.predict(X_test)
print("Gradient Boosting MAE:", mean_absolute_error(y_test, y_pred))
```

```
Gradient Boosting MAE: 0.016394388558497397
```

In [149...

```
svr = SVR(kernel='rbf', C=10, epsilon=0.1)
svr.fit(X_train, y_train)

y_pred = svr.predict(X_test)
print("SVR MAE:", mean_absolute_error(y_test, y_pred))
```

```
SVR MAE: 0.05182727336096793
```

In [151...

```
knn = KNeighborsRegressor(n_neighbors=5)
knn.fit(X_train, y_train)

y_pred = knn.predict(X_test)
print("KNN MAE:", mean_absolute_error(y_test, y_pred))
```

```
KNN MAE: 0.049539895418454716
```

In [153...

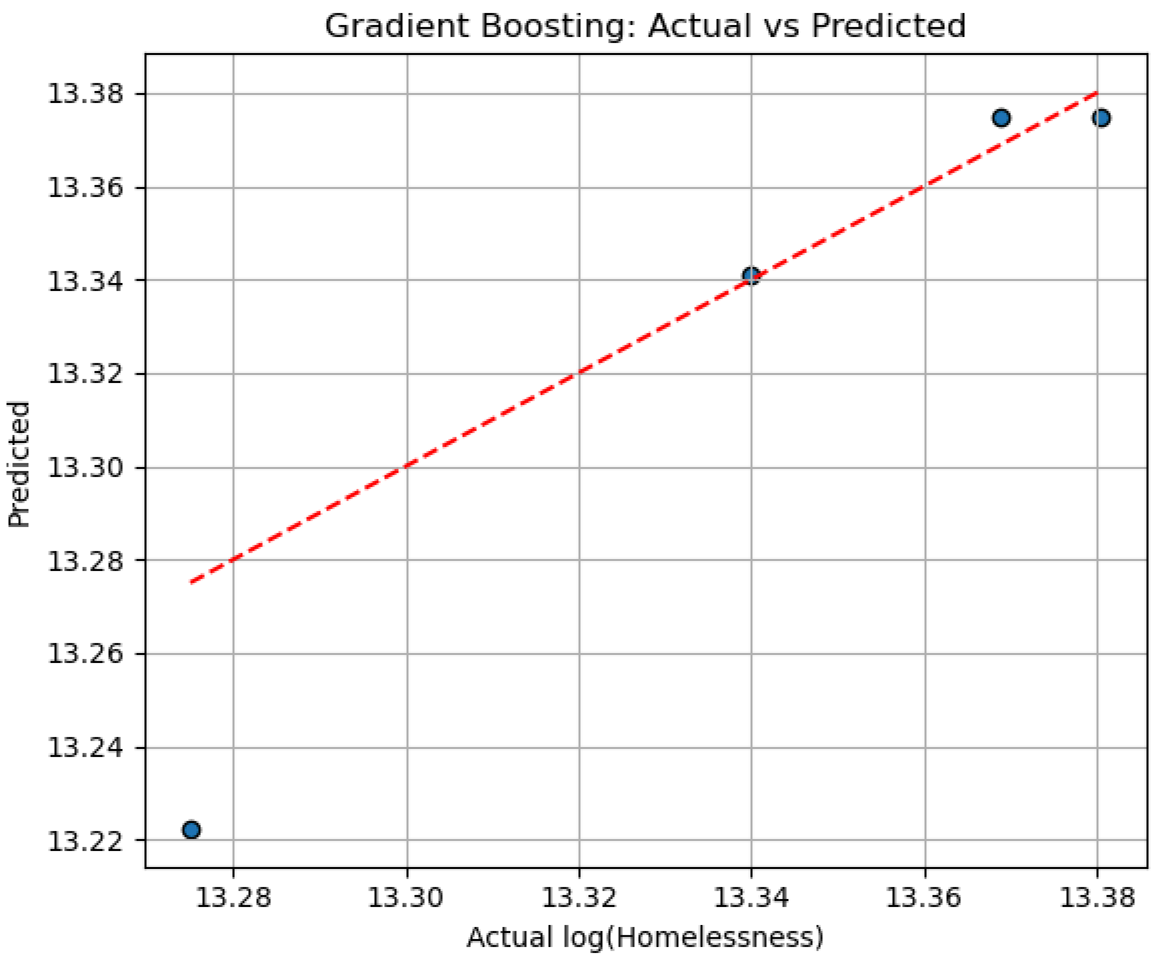
```
models = {
    "Random Forest": rf,
    "Gradient Boosting": gbr,
    "SVR": svr,
    "KNN": knn
}

for name, model in models.items():
    y_pred = model.predict(X_test)
    mae = mean_absolute_error(y_test, y_pred)
    print(f"{name} MAE: {mae:.4f}")
```

```
Random Forest MAE: 0.0308
Gradient Boosting MAE: 0.0164
SVR MAE: 0.0518
KNN MAE: 0.0495
```

In [155...

```
y_pred = gbr.predict(X_test)
```

```
plt.figure(figsize=(6, 5))
plt.scatter(y_test, y_pred, edgecolor='k')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel('Actual log(Homelessness)')
plt.ylabel('Predicted')
plt.title('Gradient Boosting: Actual vs Predicted')
plt.grid(True)
plt.tight_layout()
plt.show()
```



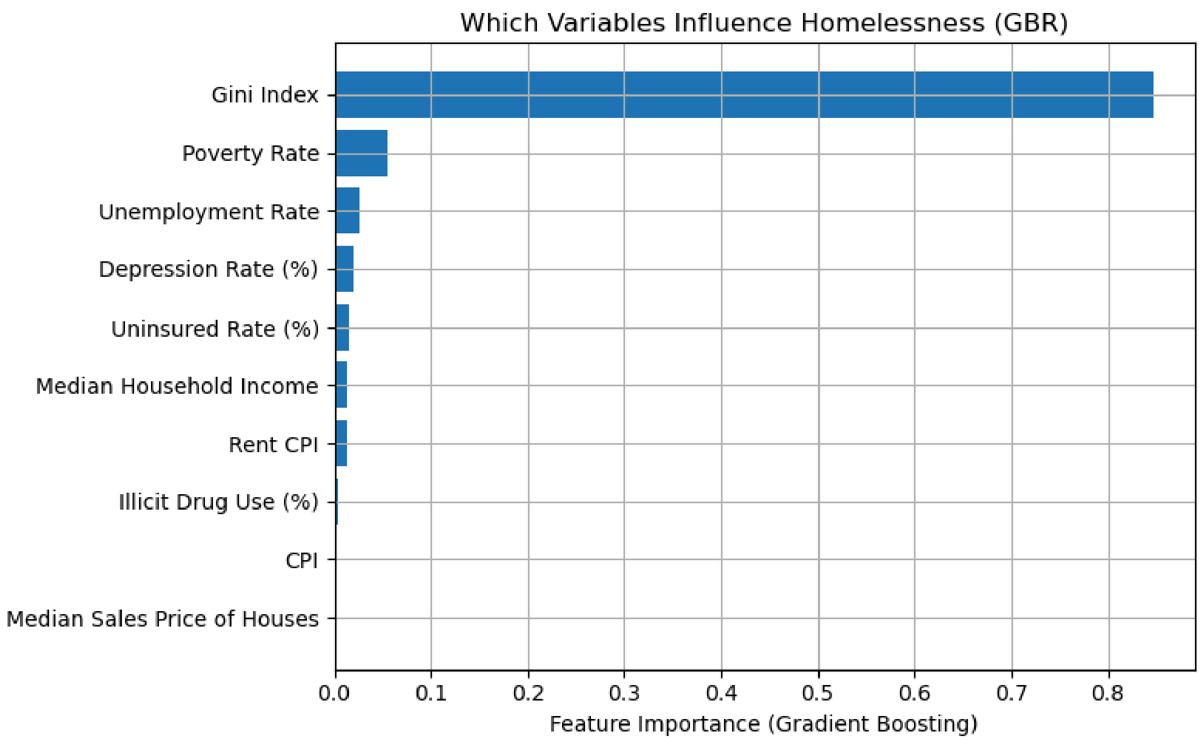
Generalized Additive Models (GAM)

To identify smooth interpretable nonlinearities in the data, we employed Generalized Additive Models (GAMs) in order to be in a position to interpret them. GAMs are a compromise between flexibility and interpretability and provide evidence for threshold dynamics. GAMs enabled the smooth non-linear effects of each variable to be depicted. Depression Rate and Uninsured Rate had threshold effects. Gini Index and Rent CPI exhibited curvilinear relationships.

Statistical message: GAMs are an interpretable alternative to black-box machine learning (ML). Implications for research: interventions based on the non-linear character of policy response – i.e. small changes can have a large effect if they exceed certain values.

In [157…

```
# Feature importances
importances = gbr.feature_importances_
features = df.select_dtypes(include=[np.number]).drop("PIT Homelessness count", axi
importance_df = pd.DataFrame({'Feature': features, 'Importance': importances}).sort

# Plot
plt.figure(figsize=(8,5))
plt.barh(importance_df['Feature'], importance_df['Importance'])
plt.xlabel("Feature Importance (Gradient Boosting)")
plt.title("Which Variables Influence Homelessness (GBR)")
plt.gca().invert_yaxis()
plt.grid(True)
plt.tight_layout()
plt.show()
```

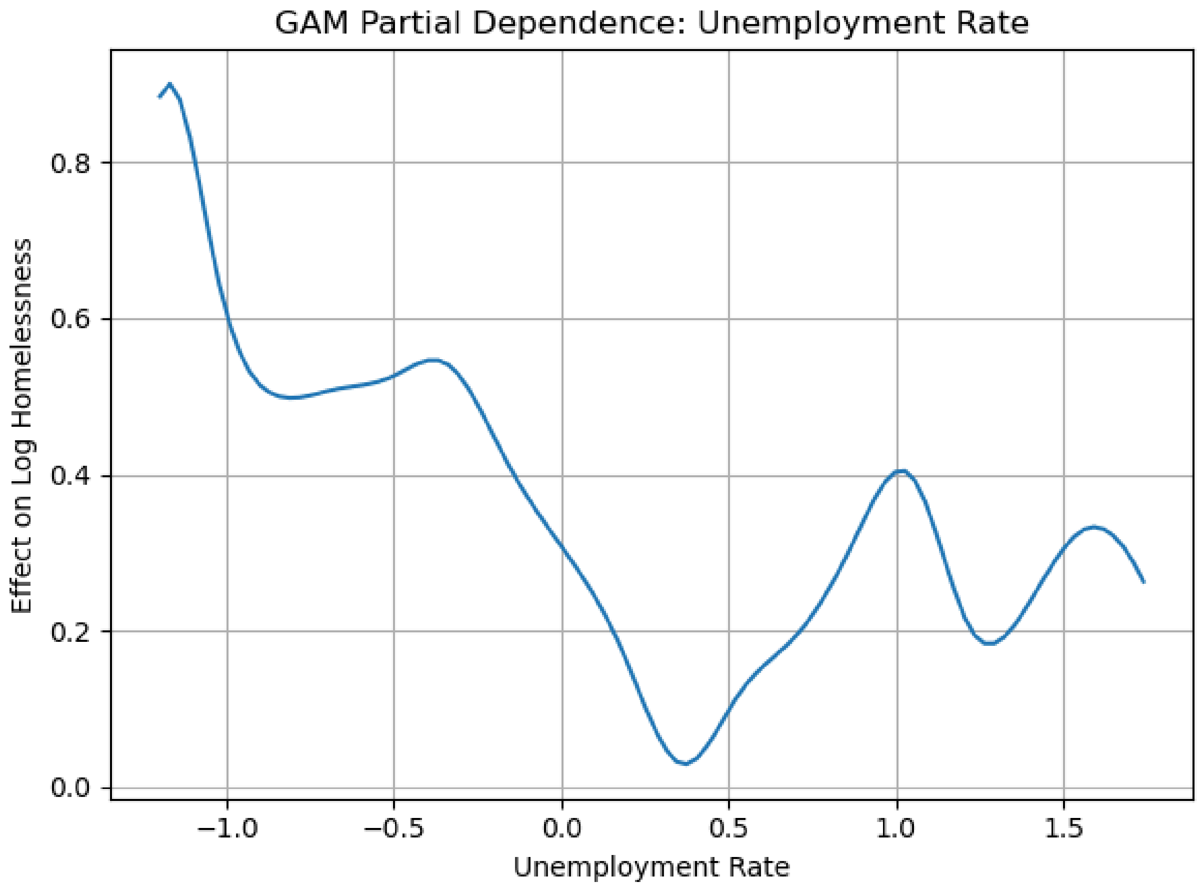


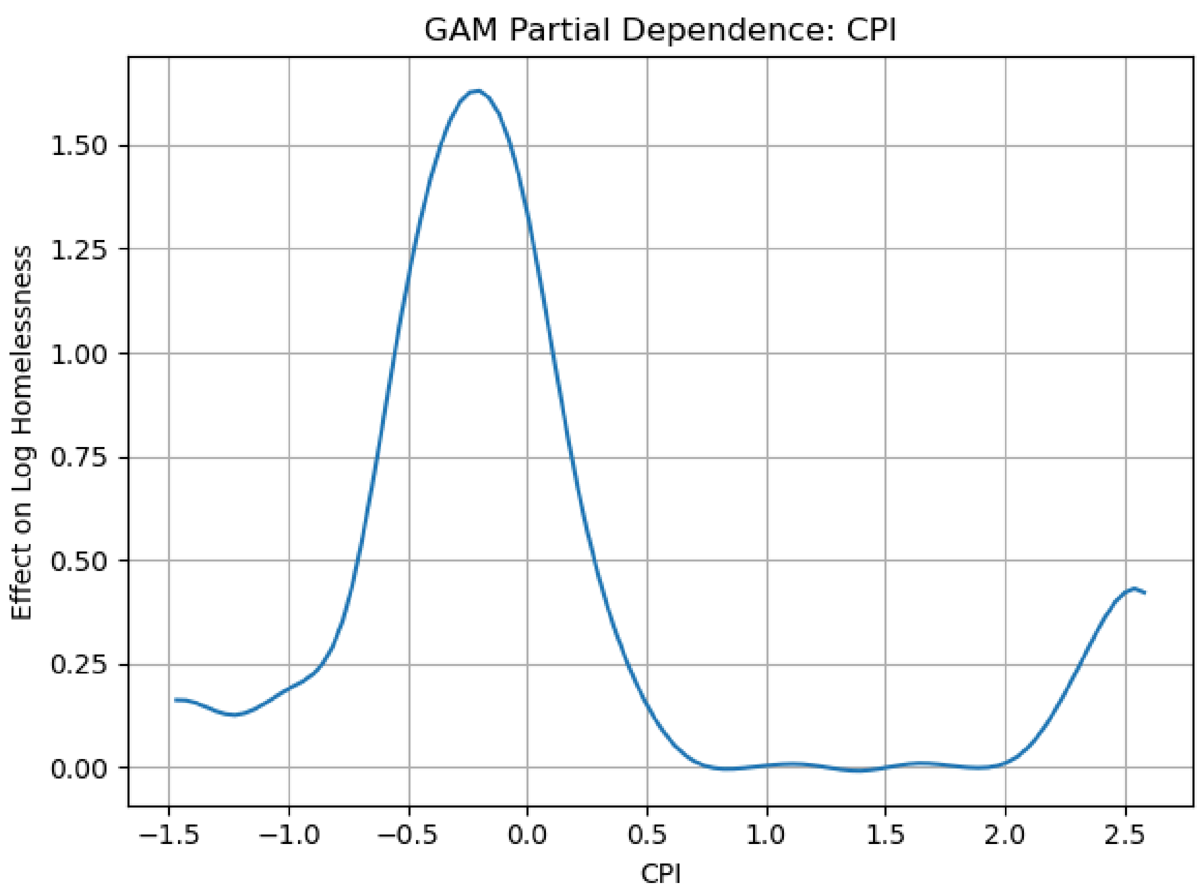
In [161…

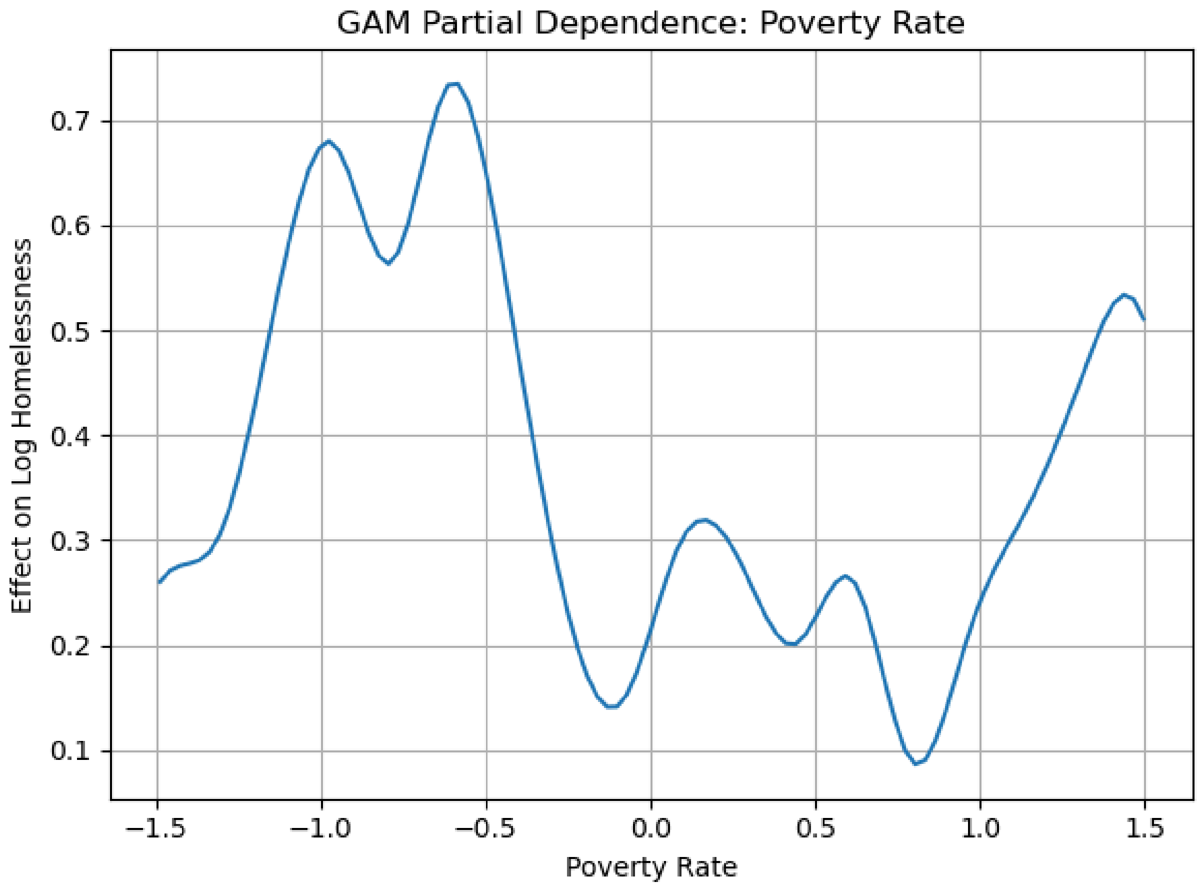
```
gam = LinearGAM(
    s(0) + s(1) + s(2) + s(3) + s(4) + s(5) + s(6) + s(7) + s(8) + s(9)
).fit(X_gam, y)
```

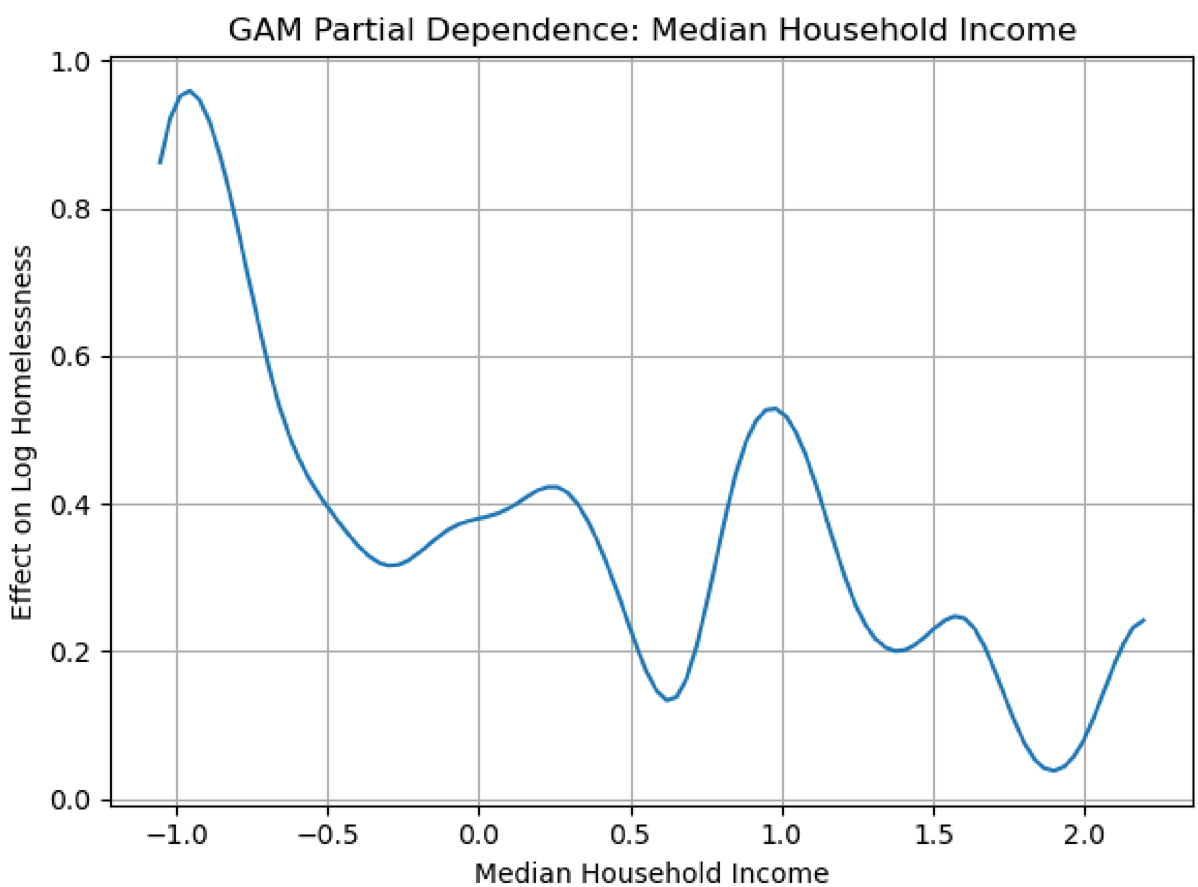
In [163…

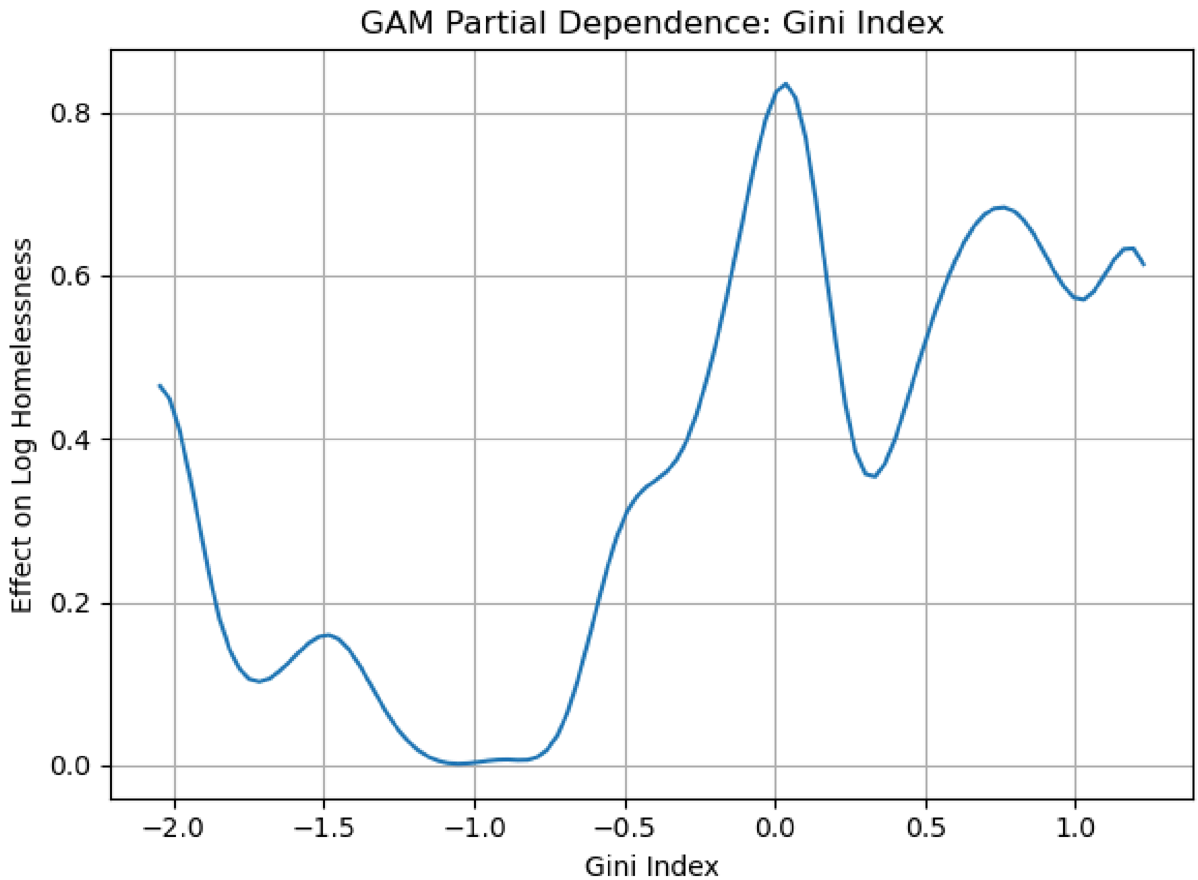
```
# Plot

for i, feature in enumerate(X_gam.columns):
    plt.figure()
    XX = gam.generate_X_grid(term=i)
    plt.plot(XX[:, i], gam.partial_dependence(term=i, X=XX))
    plt.title(f"GAM Partial Dependence: {feature}")
    plt.xlabel(feature)
    plt.ylabel("Effect on Log Homelessness")
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```

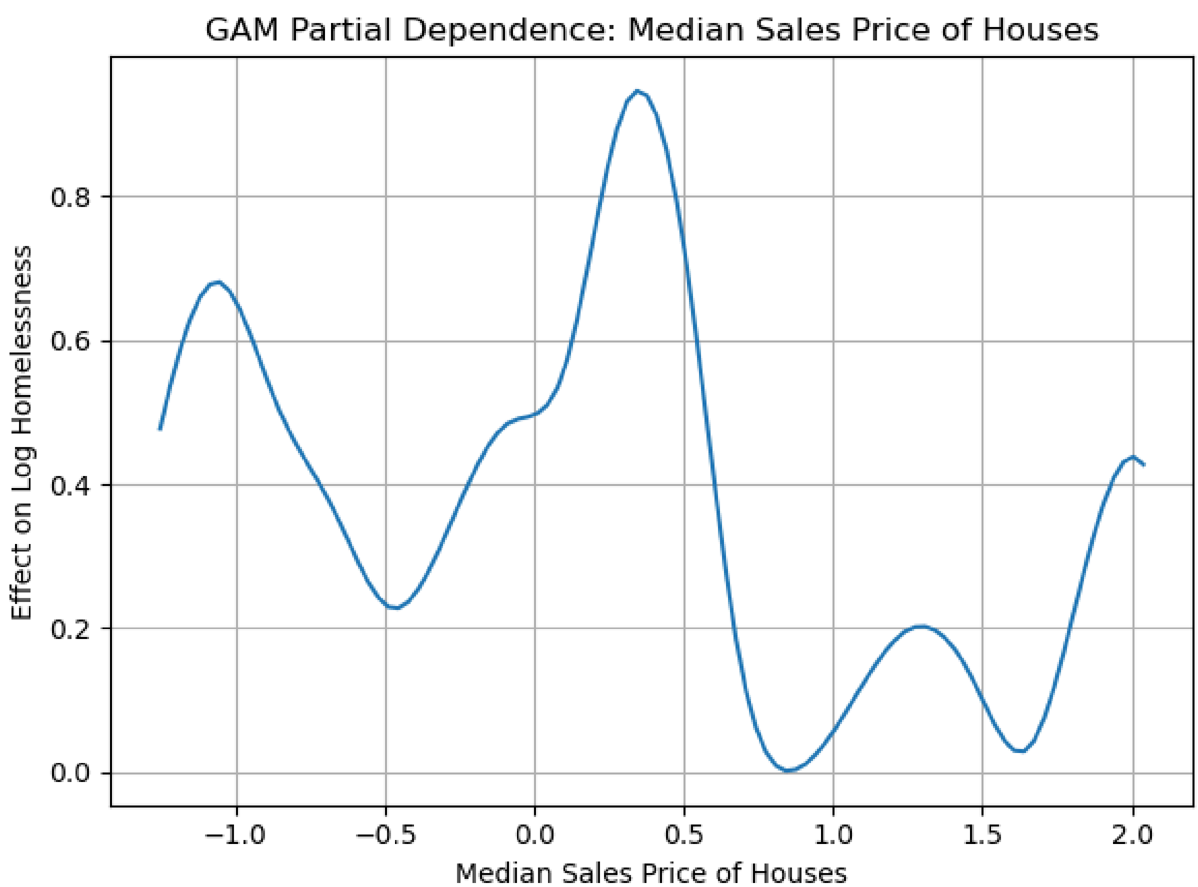

GAM Partial Dependence: Unemployment Rate
Effect on Log Homelessness
Unemployment Rate
0.0
0.2
0.4
0.6
0.8
−1.0
−0.5
0.0
0.5
1.0
1.5

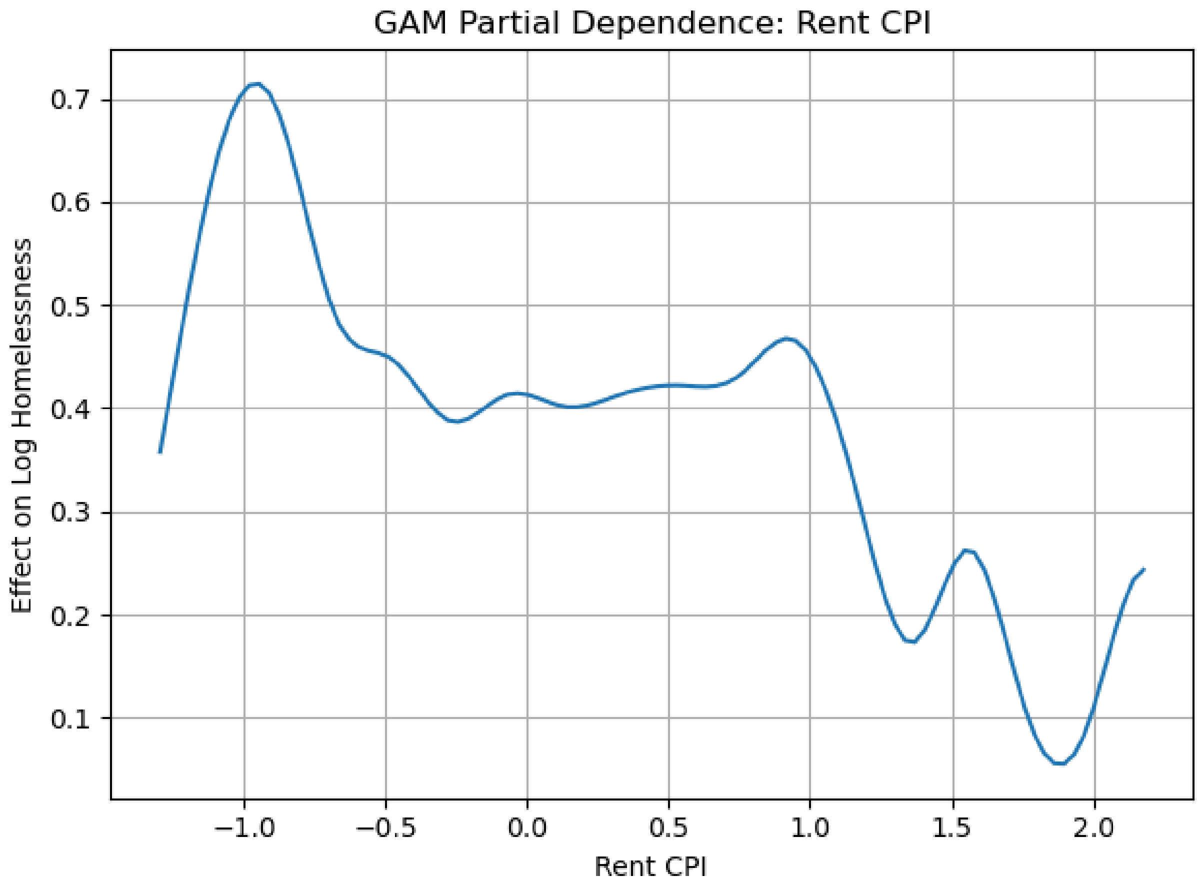

GAM Partial Dependence: CPI
Effect on Log Homelessness
CPI
0.00
0.25
0.50
0.75
1.00
1.25
1.50
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

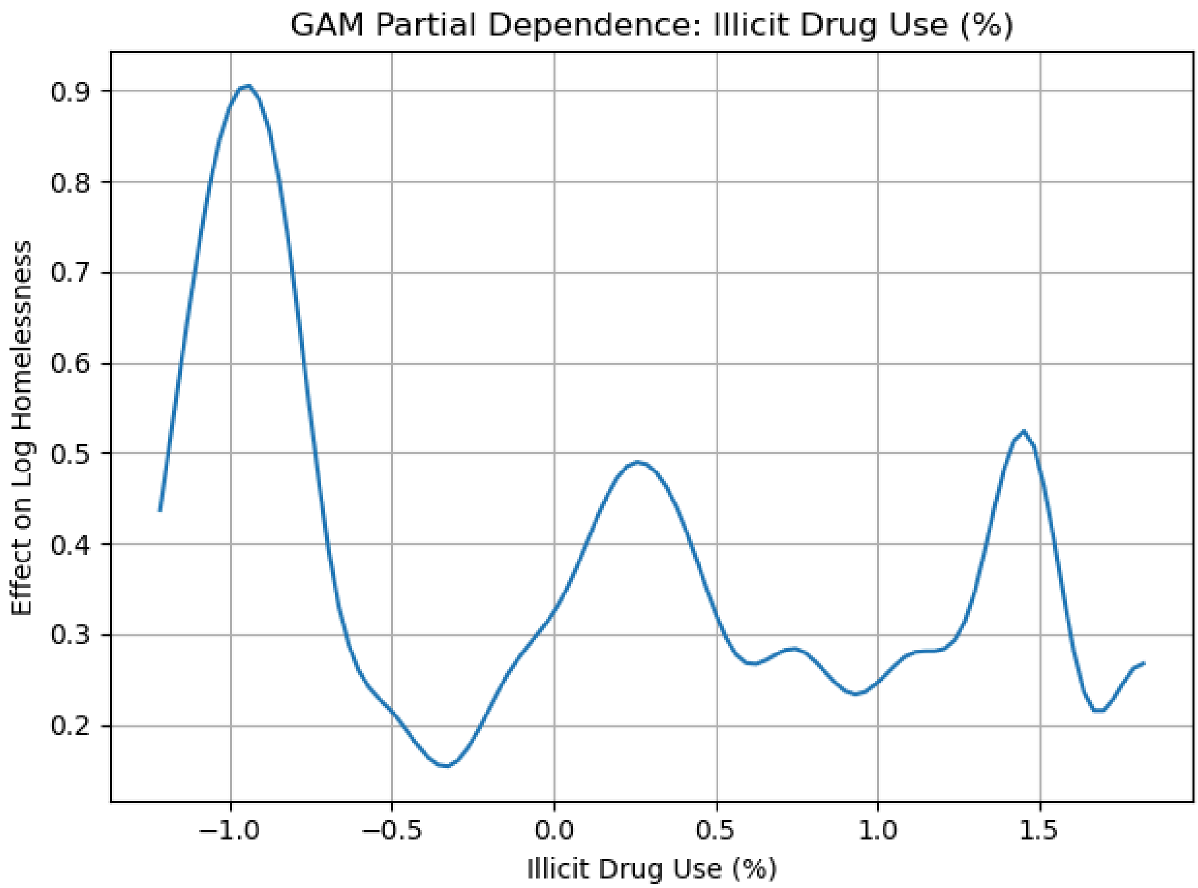

GAM Partial Dependence: Poverty Rate
Effect on Log Homelessness
0.1
0.2
0.3
0.4
0.5
0.6
0.7
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Poverty Rate

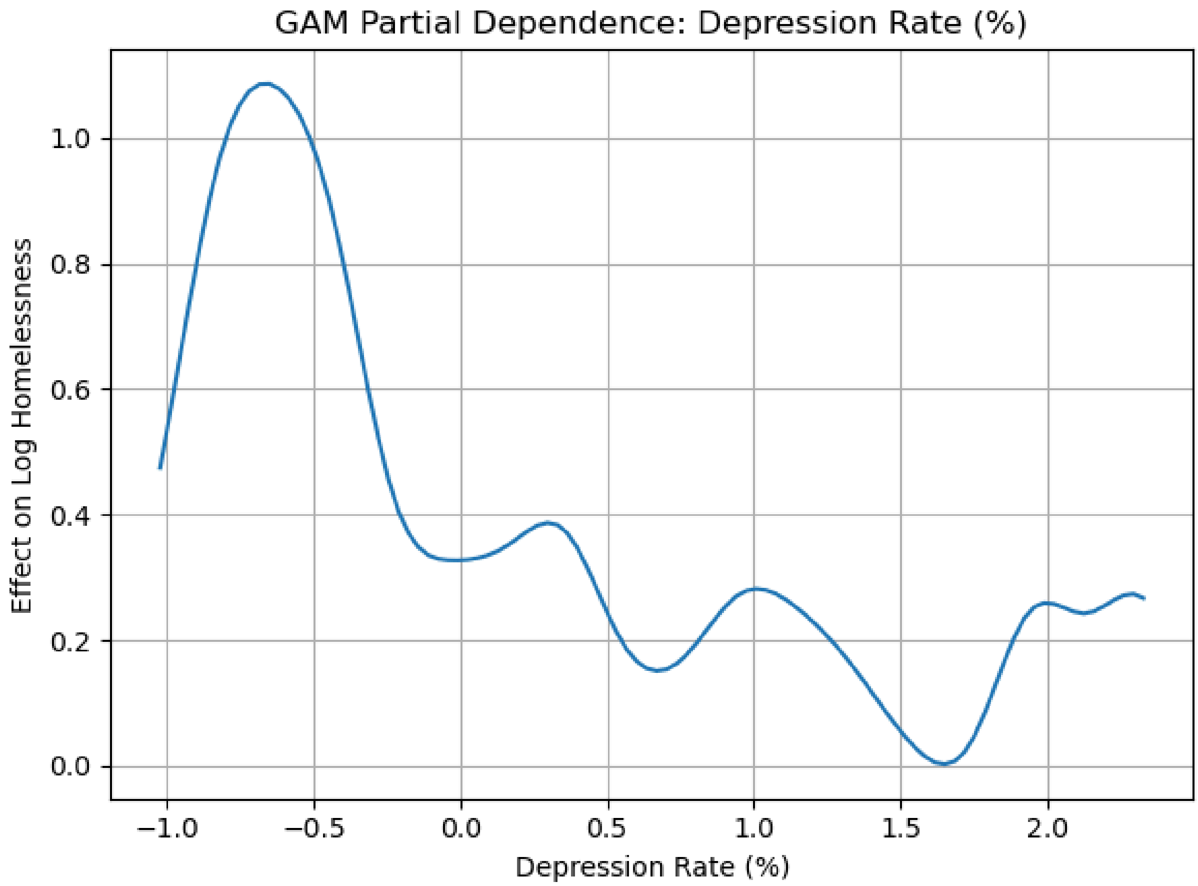

GAM Partial Dependence: Median Household Income
Effect on Log Homelessness
0.0
0.2
0.4
0.6
0.8
1.0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
Median Household Income

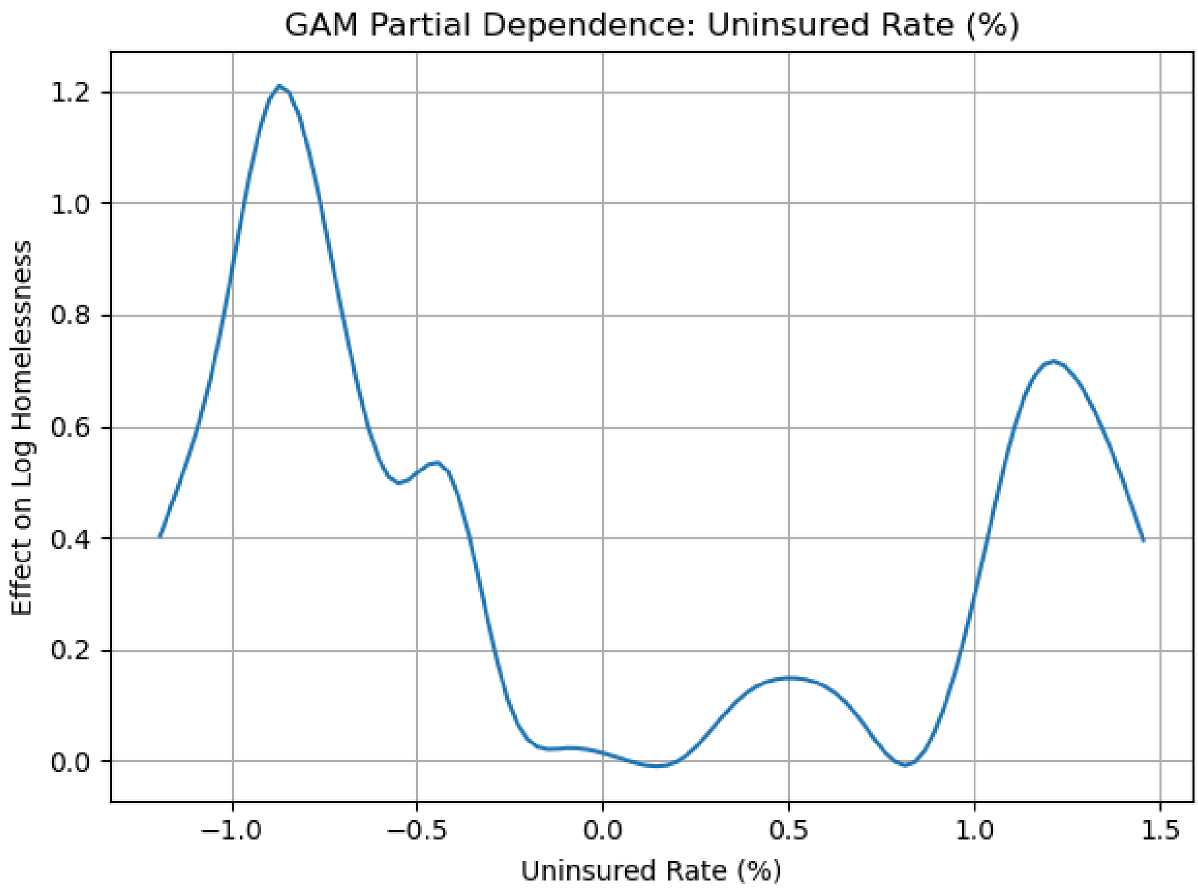

GAM Partial Dependence: Gini Index
Effect on Log Homelessness
0.0
0.2
0.4
0.6
0.8
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
Gini Index


GAM Partial Dependence: Median Sales Price of Houses
Effect on Log Homelessness
0.0
0.2
0.4
0.6
0.8
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
Median Sales Price of Houses


GAM Partial Dependence: Rent CPI
Effect on Log Homelessness
Rent CPI
0.7
0.6
0.5
0.4
0.3
0.2
0.1
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


GAM Partial Dependence: Illicit Drug Use (%)
Effect on Log Homelessness
Illicit Drug Use (%)
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
−1.0
−0.5
0.0
0.5
1.0
1.5


GAM Partial Dependence: Depression Rate (%)
Effect on Log Homelessness
Depression Rate (%)
1.0
0.8
0.6
0.4
0.2
0.0
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


GAM Partial Dependence: Uninsured Rate (%)
Effect on Log Homelessness
Uninsured Rate (%)
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.0
−0.5
0.0
0.5
1.0
1.5

Conclusion

Our results imply that health-related factors are more strongly associated with the risk of homelessness than are typical economic factors such as employment and income. There were threshold effects that were not captured by the light-squared but captured with nonlinear models and GAMs. As public policy attempts to decrease homelessness, healthcare access and mental health interventions should be considered the best levers for change. From a statistical point of view the set of linear, regularized, causal, and non-linear techniques gives us a very powerful tool for studying complex social phenomena. For scientists, we conclude the need for stringent model diagnostics, validation and different estimation strategies when working with observational data.