Housing Affordability in the United States

Is Density Truly the Biggest Predictor of Rent Prices?

Statistical Analysis Report

December 2023

Breck Emert

Stat 840

Abstract

This study investigates the relationship between rental prices and various economic factors across the United States, with a focus on the influence of population density on this dynamic. Utilizing a dataset spanning 17 years, we analyzed state-level data on rental prices, vacancy rates, population density, income, unemployment, and home age. We hypothesize that higher population density correlates more strongly with rental prices in comparison to lower-density states. The analysis was conducted using R and involved rigorous data preprocessing, exploratory data analysis, and statistical testing, including outlier analysis and automatic model selection along with other methods.

Our findings reveal a significant difference in the impact of population density on rental prices between low-density and high-density states. In high-density states, population density strongly influences rental prices, indicated by a high F-value in our regression model. This suggests that in densely populated areas, increased competition for housing may lead to higher rental prices. The study also covers the limitations of the current dataset in capturing the full spectrum of rental market dynamics across different states, suggesting the need for further research incorporating additional variables.

This research contributes to the ongoing debate on housing policies and brings light to the importance of considering population density in policy decisions related to housing and urban development.

Contents

List of Tables

List of Figures

Title

Housing Affordability in the United States: Is Density Truly the Biggest Predictor of Rent Prices?

Introduction

There are many factors that influence rental prices, all of which are debated on small and large stages. Investigating a model for rent will help us explore a commonly cited claim made in these debates, that building more rental units will lower rental prices. We hypothesize that states with a higher population density have a higher correlation of rental price with vacancy rate than states with a low population density.

The dataset spans 17 years and contains rental prices, year, vacancy rate, population density, income. Rental price serves as the response variable, and all predictor variables other than vacancy rate serve as controls to isolate the effect of vacancy rate on rental prices. We considered the following predictor variables:

1. **State**: The state the data was recorded
2. **Year**: The year the data was recorded.
3. **Vacancy Rate**: The percentage of one-bedroom rentals that are unoccupied.
4. **Population Density**: The number of people living in one square mile of land.
5. **Income**: The real median household income as measured in March.
6. **Unemployment**: The percentage of working-age individuals unemployed
7. **Structure Age**: The average age of housing and rental units

In summary, this study aims to critically analyze the relationship between vacancy rates and rental prices at the state level. By analyzing data across a range of states and years, we intend to test the hypothesis that the correlation between rental prices and vacancy rates increases in high-density states compared to low-density states. This investigation will support data-driven policy decisions in the ongoing debate on how to improve rent prices.

Materials and Methods

Data Collection

The dataset used for analysis was compiled from reliable government sources, the US Census Bureau and the Office of Policy Development and Research, and spans 17 consecutive years of state-level data from 2006 to 2022. The scope of this analysis was intentionally confined to the contiguous United States to maintain homogeneity in the housing market dynamics. Alaska and Hawaii were excluded due to their distinctive geographic, demographic, and economic characteristics. Each record within the dataset represents an individual state's yearly data, which gives a large range of values across each 50 states. Several ranges of economic cycles and housing market trends are present in the data, which gives diversity to the dataset for our investigation.

Data Preprocessing

Prior to analysis, the dataset was inspected for appropriateness. This stage involved identifying and addressing missing/duplicate values, datatypes, and formats. The dataset was found to be appropriate and did not require any non-linear modification to the values. For clarity in visualization and interpretation, income values were scaled down by a factor of 1,000, thus representing income in thousands. This adjustment lowers the large number of digits in results and does not alter the relationships between variables.

Statistical Analysis

The data analysis was performed with the statistical language R inside of R Markdown. Data was loaded from an Excel format into R dataframes, with our code visible in the Appendix. The variables were explored individually for appropriateness and errors; no missing values were found in the dataset. Subgroups, termed "low-density" and "high-density" states, were created. States were sorted into these subgroups by comparing their population density against the median population density threshold. A general linear model was selected to model the data, and automatic model selection methods were used to aid the final model selection. The model assumptions were assessed and confirmed. All statistical tests will be conducted with a significance level of 0.05.

Final Dataset

The names of the variables will hereby be referred to by their name in the dataset, for consistency, as visible in the data table below.

Table 1: Final Dataset

Variable Name	Data Type	Data Format	Description	Source	Example
State	Categorical	Text	The State within the contiguous U.S.		Kansas
Year	Numeric	YYYY	Year to which the data pertains		2023
Rent	Numeric	#,###	40th percentile rent for 1-bedroom apartments	Office of Policy Development and Research	$1,550
Density	Numeric	#,###	Population per square mile of land	U.S. Census Bureau	280
StructureAge	Numeric	#,###	The average year built of housing and rental units	U.S. Census Bureau	1971
Income	Numeric	###.#	Household median income, in thousands	U.S. Census Bureau	$55k
Vacancy	Percentage	0.00%	Percentage of vacant rentals	U.S. Census Bureau	4.40%
Employment	Percentage	0.00%	The percentage of working-age individuals exmployed	Bureau of Labor Statistics	2.5%

Results

Exploratory Data Analysis

The first step of our exploratory phase was to build a comprehensive table of the dataset's descriptive statistics (excluding Year). The average rent of the dataset was $636 and ranges from $387 in South Dakota to $1325 in Massachusetts. Density averages 203 people per square mile and ranges from 5.38 in Wyoming to 1261 in New Jersey. Density and Rent show strong skewness, at 2.24 and 1.28 respectively, while Employment and Vacancy show moderate skewness.

Table 2: Descriptive Statistics for the Dataset

	mean	median	sd	min	max	skewness
Year	2014	2014	4.90	2006	2022	0.00
Vacancy	8.16	7.75	2.96	2.40	18.10	0.55
Density	202.71	102.86	266.36	5.38	1260.61	2.24
Income	69.32	68	11.81	40	109	0.47
Employment	5.63	5.10	2.22	2.07	13.73	0.86
Rent	636.12	590	165.69	387	1325	1.28

The histograms of the variables show this data visually, and highlight the strong skews.

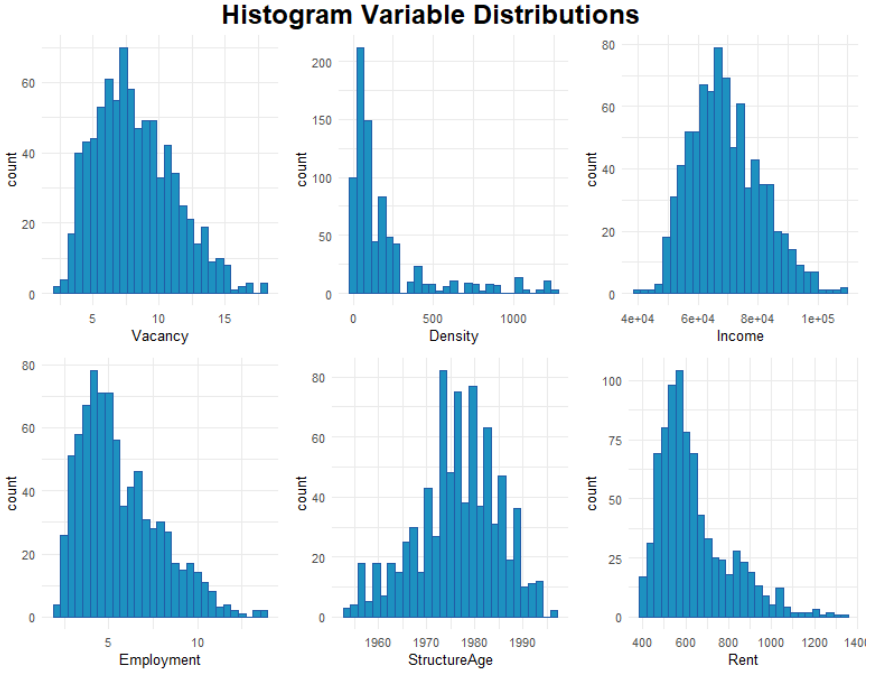

Figure 1: Histogram Variable Distributions

Log-transformations proved to significantly reduce the skew for many variables, and the distributions showed log-normal distributions. For log-log relationships in a model, the coefficient is a close estimation of the percentage change in Y for a percentage change in X (Duke, n.d.). The necessity for log transformation, and the respective descriptive statistics table, was further examined in Appendix A.

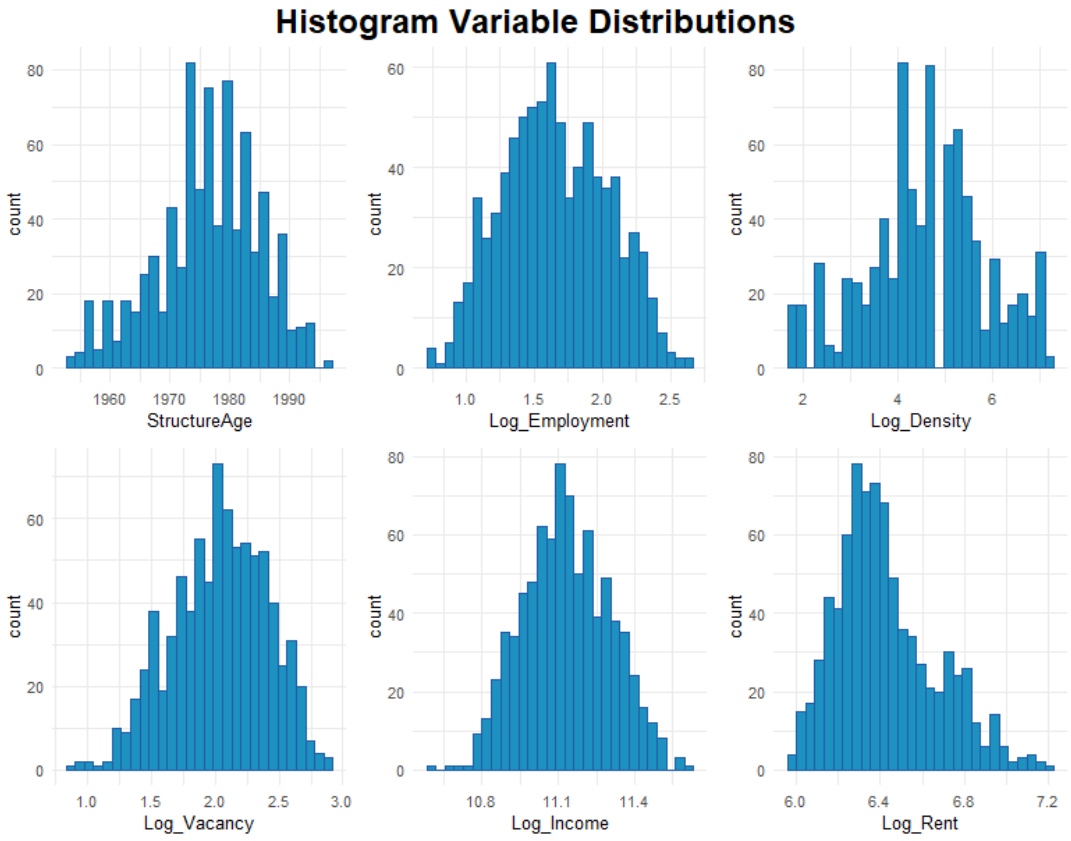

Figure 2: Log-Transformed Histogram Variable Distributions

With individual distributions accounted for, the bivariate distributions were visualized with a scatterplot matrix. The response variable is located on the bottom row and rightmost column of the plot. Log_Rent vs. Log_Density shows potential for a non-linear relationship, but the plot otherwise shows linear trends between the response and the predictors. Upon further testing through Component-Residual plots and normality testing, there was not sufficient evidence to warrant inclusion of non-linear predictor for Log_Rent vs. Log_Density.

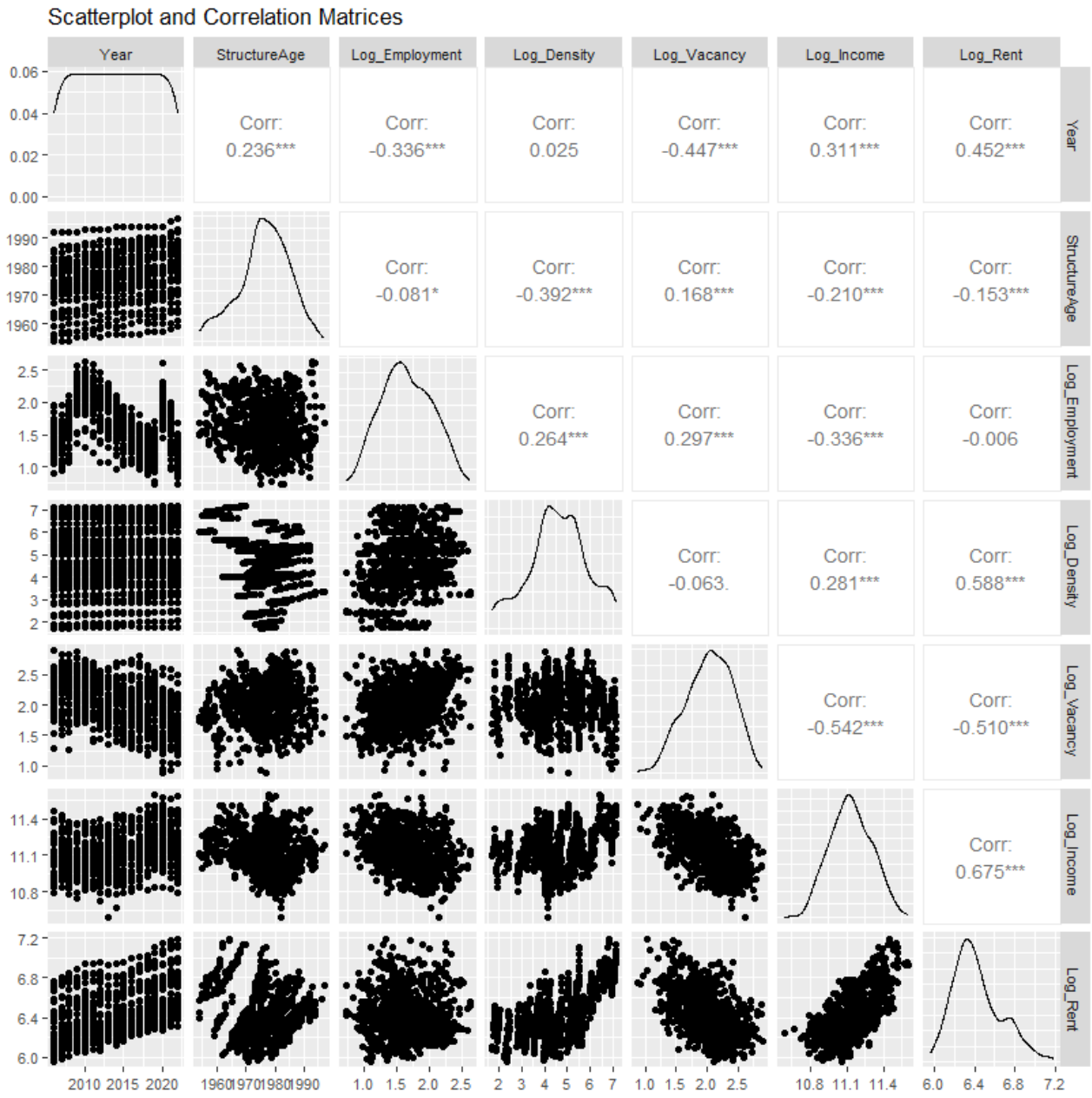

Figure 3: Scatterplot and Correlation Matrix

Correlation Analysis

Here, the method of hypothesis testing was set up by splitting the data into low- and high-density, determined by the row's population density relative to the median. The correlation matrices comparing low- and high-density entries shows many large changes between the correlations, with the largest being the correlation of interest, Log_Density with Log_Rent, going from 0.05 to 0.76.

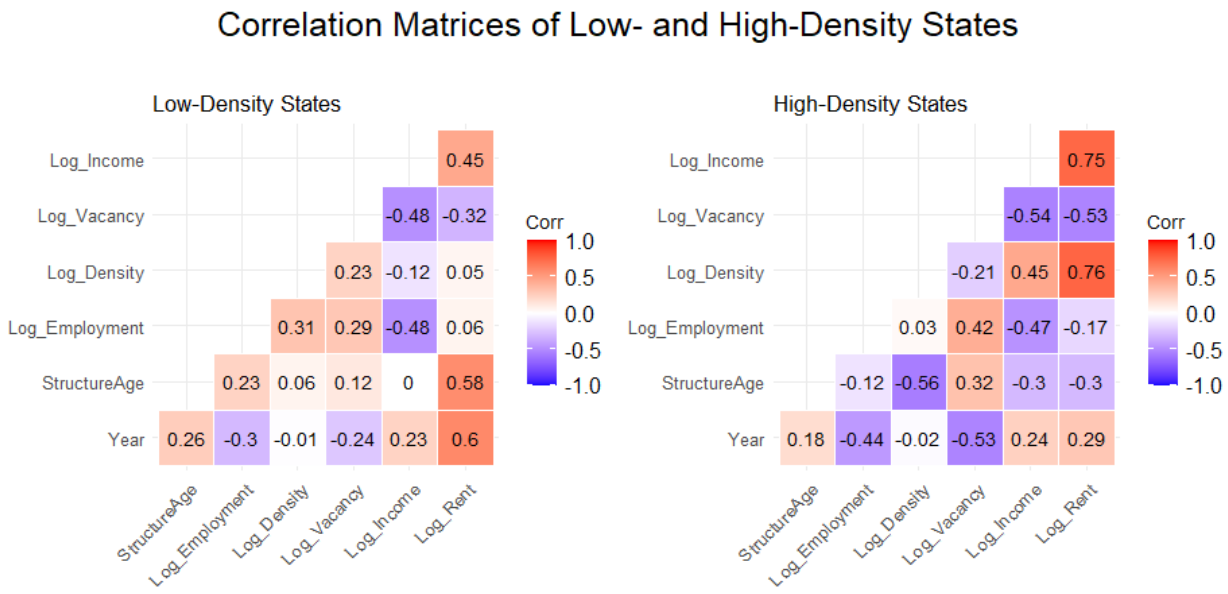

Figure 4: Correlation Matrices of Low- and High-Density States

Outlier Analysis

Log-transformations significantly improved distribution normality, but outliers remained in the histograms and scatterplots. Most prominent are outliers in Log_Vacancy (notably Vermont's vacancy rate in 2006 and 2008) and in Log_Income (notably Mississippi's median income in 2013 and 2014). These data points show as visual outliers but do not exceed 1.5 times the interquartile range (a common threshold), nor were explanations found to justify their removal (Vermont Housing Financing Agency, 2008). Reviewing the full records associated with these outliers shows no substantial deviation across other variables. Given the dataset's size (817 observations), the descriptive statistics remained nearly unchanged upon removal of these 4 outliers, and therefore were not removed. All observations noted here were carefully considered during the leverage analysis of the full model.

Covid-19 (2020-2022) corresponds to a large spike in unemployment, a reversal of the trend in income, and increased variance in vacancy rates. Because of the economic significance of this event and its clear impact on the data, the final years of the dataset, 2020-2022, were removed.

Outliers visible in the scatterplots corresponded with the most extreme residuals of a strong candidate model. Certain states, particularly Vermont and Louisiana, show extreme residuals, and make up most of the outlying leverage points. They exhibit extreme effects on the regression coefficients.

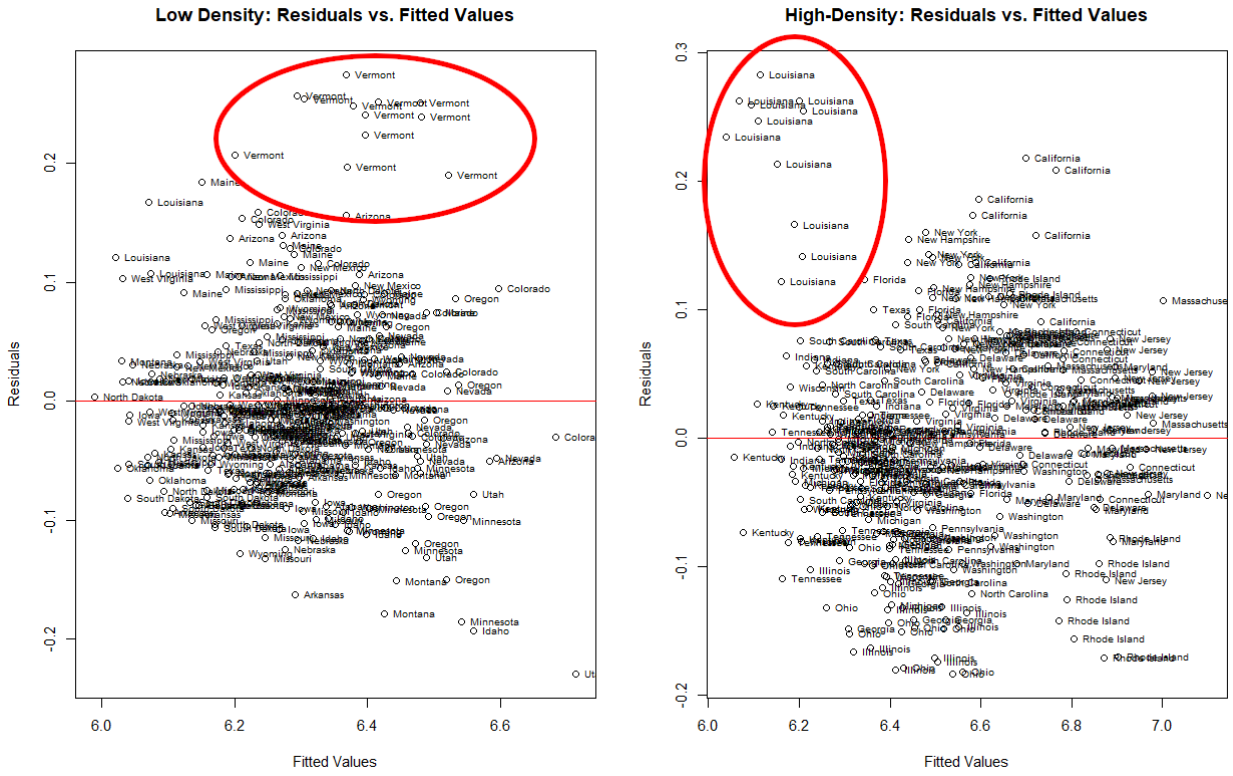

Figure 5: Residuals Highlighting State Outliers

The removal of these outliers was carefully considered and further justified by the Shapiro-Wilk test, which indicated a thousand-fold improvement in the residual normality upon their removal. Maine was identified as an outlier through similar diagnostics, and boosted the low-density model normality by another thousand-fold to achieve normality of the residuals. These removals significantly limit the study's scope but were necessary to maintain model assumptions and capture generalizable trends with the given variables. Possible explanations will be examined in the study limitations sections, and may benefit from the inclusion of more variables.

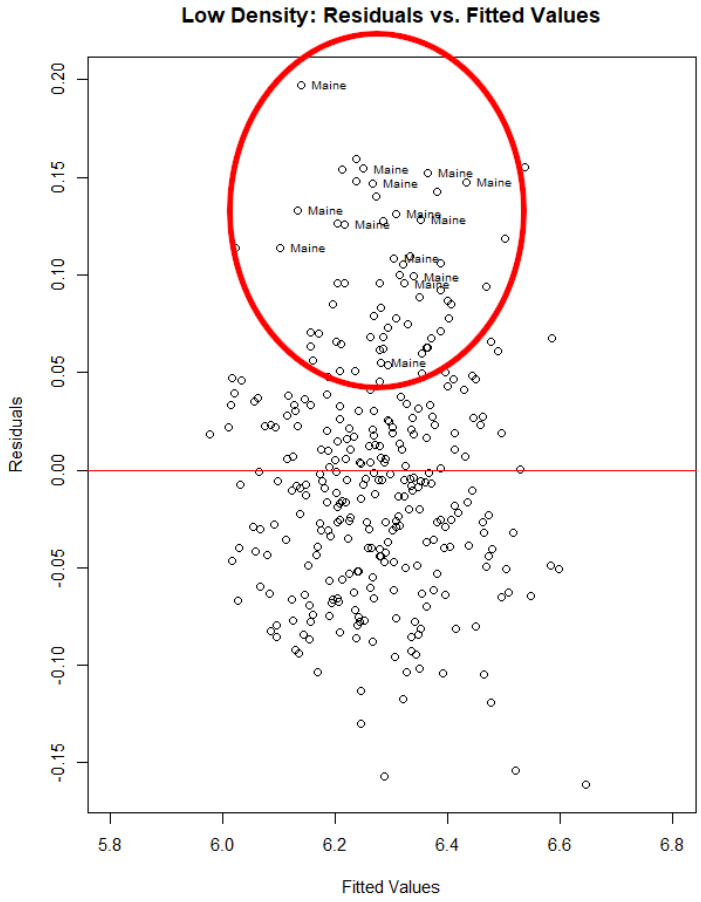

Figure 6: Residuals Highlighting Maine Outliers

Primary Objective Analysis

Variable Selection

Prior to testing for the best model, we evaluated potential issues with each variable, including multicollinearity and theoretical relevance, on top of previous observations. A basic component-residual plot highlights whether a linear fit for the excluded variable best explains the trends left unexplained by the model. In response to the non-linearity observed in the plot, a square term will be added for Log_Vacancy, allowing the linear model to fit the trend. The lack of trend in Log_Density may suggest it does not add value to the model's explanatory power, but because of the moderate size of the dataset, may still hold relevance. Both variables will be analyzed for significance and explanatory power during model testing.

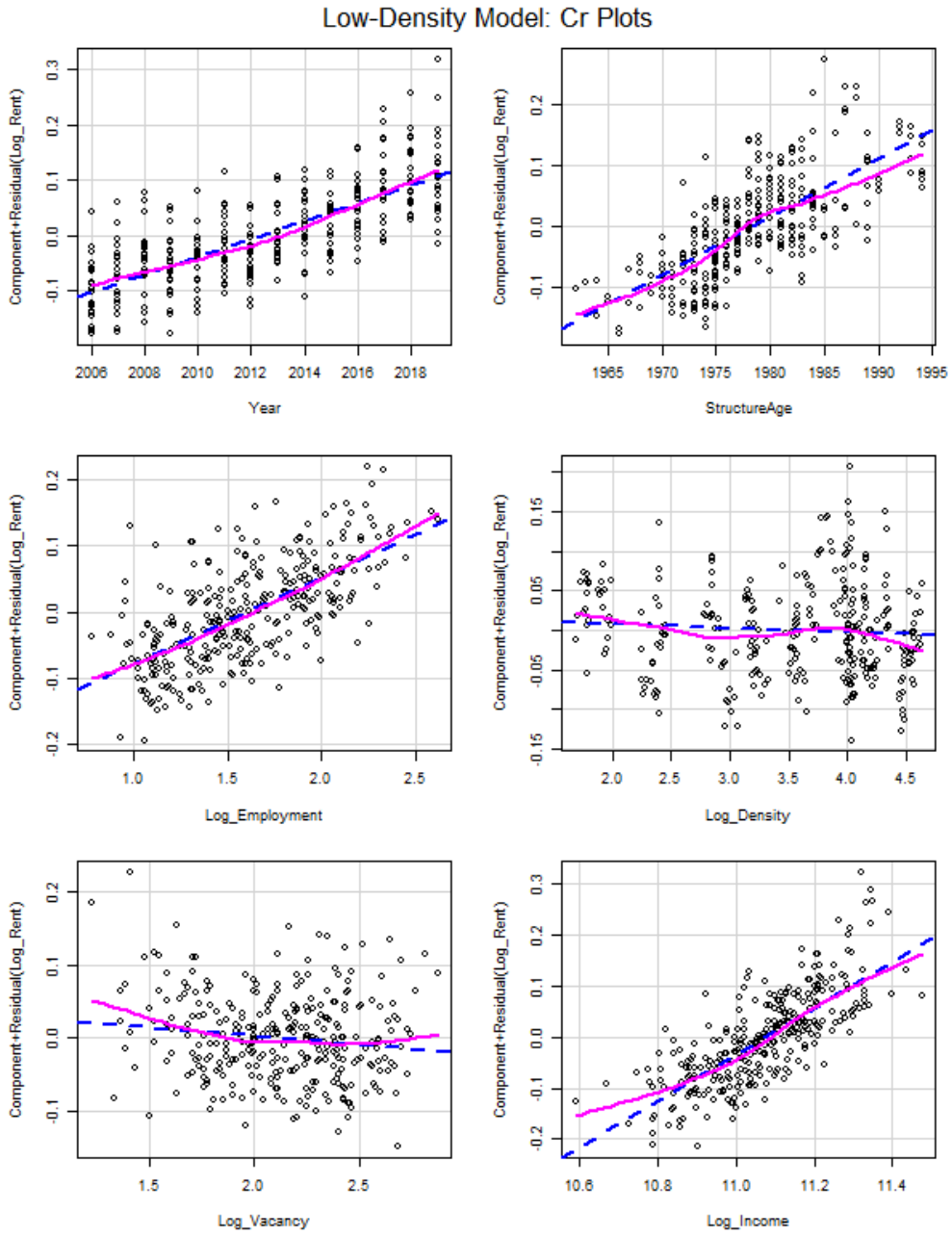

Figure 7: Low-Density Model Component-Residual Plots

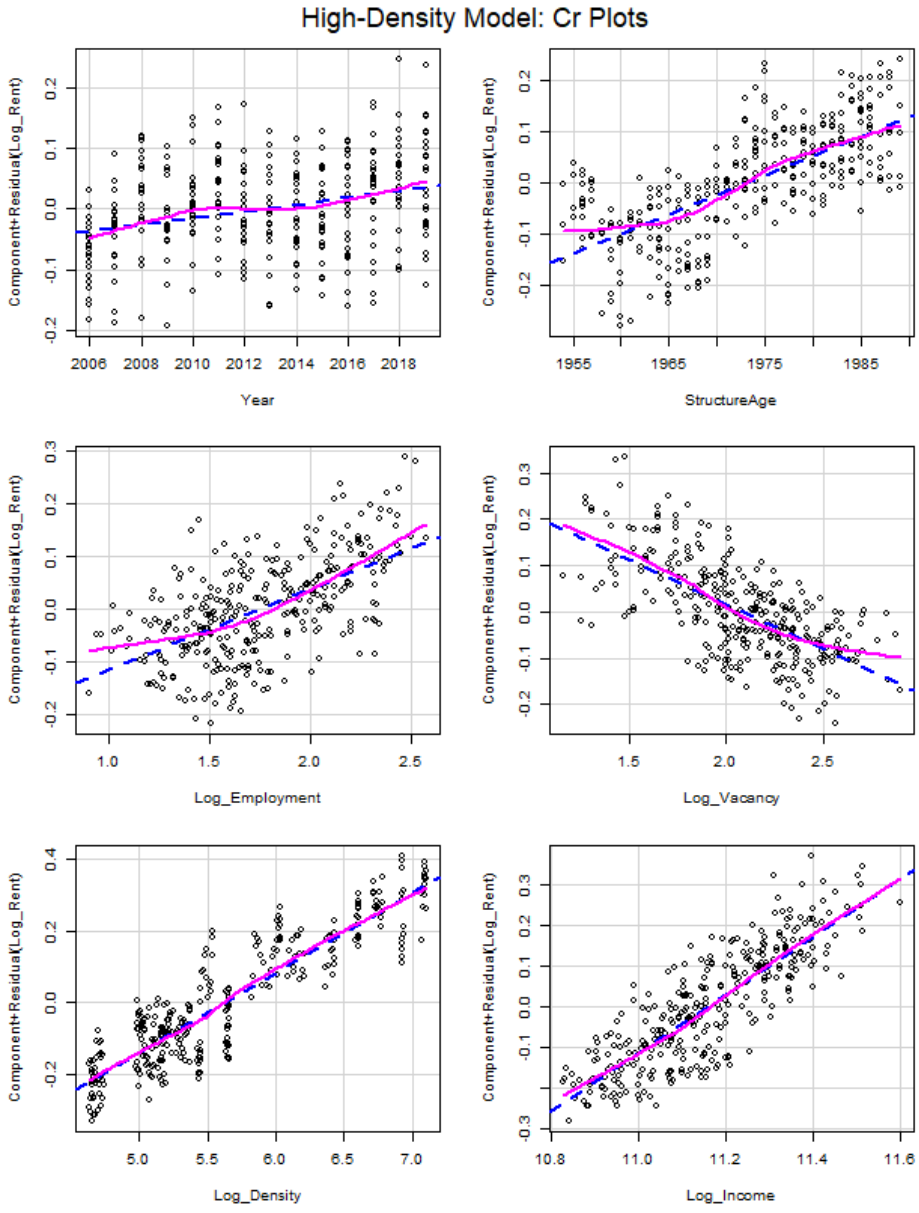

Figure 8: High-Density Component-Residual Plots

Inflation of the model coefficient's variance occurs with covariance, and is quantified using the Variance Inflation Factor (VIF). The VIFs listed in Table _ indicate the factor that the variance increases due to interactions with other variables. Log_Vacancy_sq is a linear transformation of Log_Vacancy to act as a quadratic predictor, and the two are expected to have a high VIF. Given the exceptionally low VIFs otherwise, no variables were removed due to covariance concerns. The correlation matrices show a high potential for inclusion of each variable making each variable relevant for explaining the variance in marketshare, so no variables were removed.

Table 3: Variance Inflation Factors for Low- and High-Density Variables

	Year	Structure Age	Log Employment	Log Density	Log Income	Log Vacancy	Log Square Vacancy
Low-Density	1.30	1.31	1.73	1.20	1.68	123.06	123.61
High-Density	1.81	1.92	1.67	1.77	2.28	2.07	

Model Selection

To aid the selection of the final model, candidate models were found through the Akaike Information Criterion (AIC). Other automatic model selection methods including Mallow's Cp were used as supporting methods, with their outputs visible in Appendix B. The methods consistently identified the top models. Inclusion of every variable is recommended by AIC for both the low- and high-density models, but the marginal differences are low.

Table 4: Akaike Information Criterion for Model Selection

Low-Density Included Variables							size	AIC
Log_Income	**Log_Density**	**Log_Vacancy**	**Log_Vacancy_sq**	**Log_Employment**	**Year**	**StructureAge**	**7**	**-866.6**
Log_Income	Log_Vacancy	Log_Vacancy_sq	Log_Employment	Year	StructureAge		6	-866.19
Log_Income	Log_Vacancy	Log_Employment	Year	StructureAge			5	-857.4
Log_Income	Log_Density	Log_Vacancy	Log_Employment	Year	StructureAge		6	-856.74
Log_Income	Log_Vacancy_sq	Log_Employment	Year	StructureAge			5	-855.87

High-Density Included Variables							size	AIC
Log_Income	**Log_Density**	**Log_Vacancy**	**Log_Employment**	**Year**	**StructureAge**		**6**	**-722.89**
Log_Income	Log_Density	Log_Vacancy	Log_Employment	StructureAge			5	-710.64
Log_Income	Log_Density	Log_Vacancy	Year	StructureAge			5	-635.79
Log_Income	Log_Density	Log_Vacancy	StructureAge				4	-635.76
Log_Income	Log_Density	Log_Employment	Year	StructureAge			5	-631.81

We then tested the model assumptions visually and statistically, which are the assumption of homoscedasticity, constant variance of the residuals, and that the data is best explained by a linear model. The Quantile-Quantile plot and the Shapiro-Wilk test can confirm these assumptions for each model. For the Low-Density model, the Shapiro-Wilk test returns a W-value of 0.994 with an associated p-value of 0.275, indicating no significant deviation from normality. The High-Density model returns a W-value of 0.994 and a p-value of 0.285, also suggesting that the model does not significantly differ from normal.

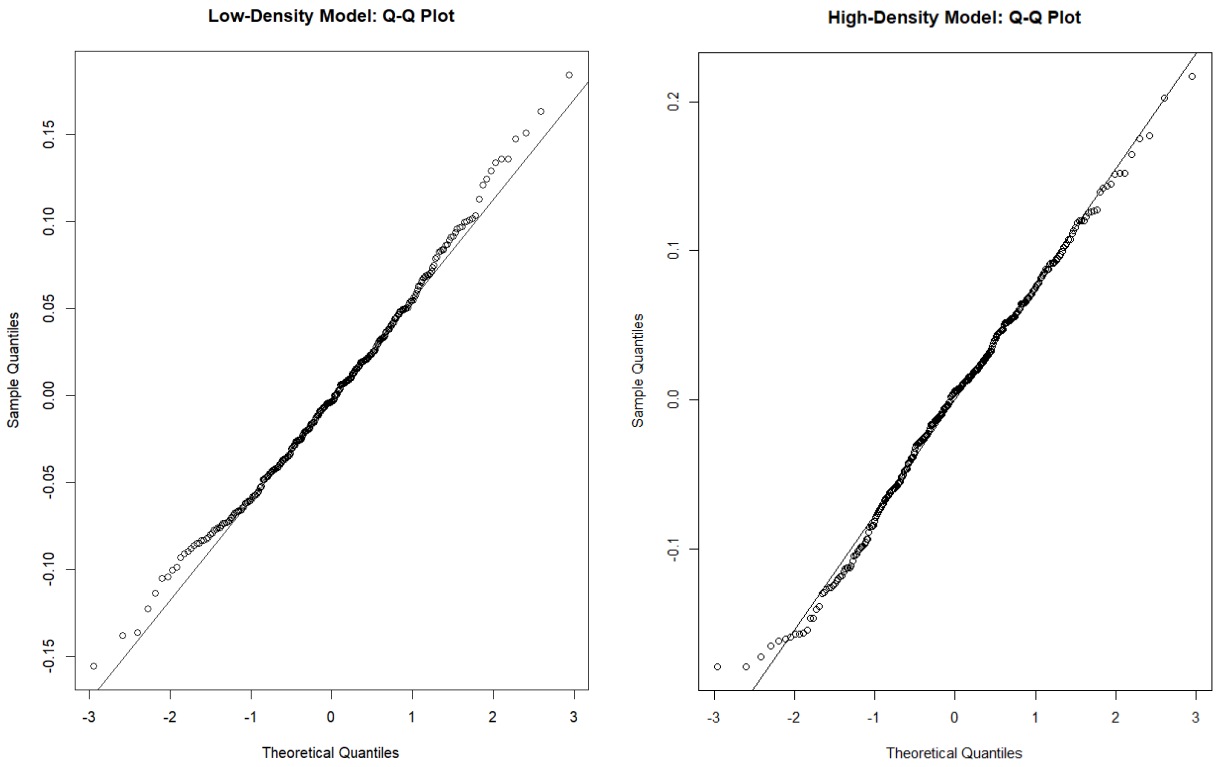

Figure 9: Quantile-Quantile plots

Both models show an even distribution of positive and negative residuals, as well as no bias towards low or high predicted values. However, there may be a slight cone-shape in the residuals vs. fitted values plot for the high-density model, possibly violating the constant variance assumption of the model. The assumption of constancy of the error variances was tested with the Breusch-Pagan test, resulting in a BP statistic of 2.571 and an associated p-value of 0.109, indicating there is not enough evidence to reject the null hypothesis that the residuals have constant variance.

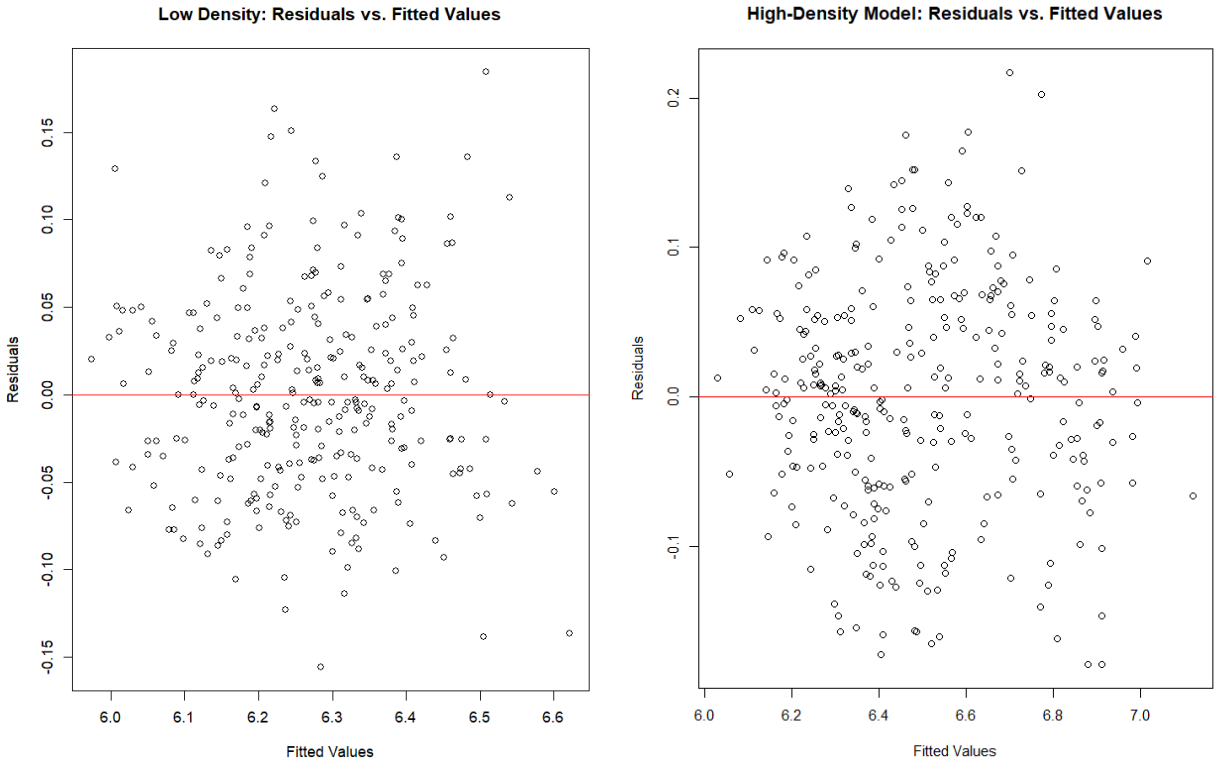

Figure 10: Residuals vs. Fitted Values

Model Validation

To evaluate the bias-variance tradeoff and predictive capability of the model, cross-validation was performed. First, Leave-One-Out Cross-Validation was performed to gauge the model's predictive power with minimal bias. This was compared to 10-fold cross validation, which gives a more balanced view of the model's performance. There is close alignment between the two tests, with minimal loss of predictive power as more of the dataset is left out of training folds.

Table 5: Cross-Validation of the Final Model

	Leave-One-Out				10-Fold		
	RMSE	**R-squared**	**MAE**		**RMSE**	**R-squared**	**MAE**
Low-Density	0.0593	0.816	0.0471		0.0586	0.828	0.0469
High-Density	0.0785	0.896	0.0627		0.0779	0.899	0.0627

To assess the incremental value of the chosen full model over simpler models, a comparative ANOVA was performed against several simpler models. Most notably, the low-density model without the quadratic predictor, and the high-density model without the Log_Employment.
To assess the incremental value of the chosen full model over simpler models, a comparative ANOVA was performed against several simpler models. Most notably, the low-density model without Log_Density, the low-density model without the quadratic Vacancy predictor, and the high-density model without Year. Both tests of low-density model options suggest that Log_Density does not significantly contribute to the model, with a low F-value of 2.35 and p-value of 0.13 in ANOVA test for removal. The marginal sum of square increase of the residuals is 0.0366, showing the negligible impact of removal on the model's explanatory power.

Table 6: ANOVA Table for Low-Density Model Comparison

ANOVA for low-density model excluding Log_Density

	Df	Sum Sq	F Value	Pr(>F)
Year	1	0.9800	286.85	$<10^{-16}$
StructureAge	1	0.7539	220.65	$<10^{-16}$
Log_Employment	1	0.4360	127.62	$<10^{-16}$
Log_Vacancy	1	0.0448	13.11	0.00034
Log_Vacancy_sq	1	0.0401	11.73	0.00070
Log_Income	1	0.8478	248.13	$<10^{-16}$
Log_Density	1	0.0080	2.35	0.13
Residuals	301	1.0284		

ANOVA for low-density model excluding Log_Vacancy_Sq and Log_Density

	Df	Sum Sq	F Value	Pr(>F)
Year	1	0.9800	277.90	$<10^{-16}$
StructureAge	1	0.7539	213.76	$<10^{-16}$
Log_Employment	1	0.4360	123.64	$<10^{-16}$
Log_Vacancy	1	0.0448	12.70	0.00042
Log_Vacancy_sq	1	0.0401	11.37	0.00085
Log_Income	1	0.8478	240.39	$<10^{-16}$
Log_Density	1	0.0080	2.28	0.13
Residuals	302	1.0650		

The high-density ANOVA test of the full model vs. a simpler model without the predictor variable Year demonstrates that each variable has significance to the model, with the Year variable having an F value of 4.01 and a p-value of 0.0462, significant at the 95% level.

Table 7: ANOVA Table for High-Density Model Comparison

ANOVA for model excluding Year

	Df	Sum Sq	F Value	Pr(>F)
Year	1	0.0860	4.01	0.0462
StructureAge	1	0.9019	42.01	<0.0001
Log_Employment	1	0.6058	28.22	<0.0001
Log_Vacancy	1	0.6370	29.67	<0.0001
Log_Density	1	4.5776	213.24	$<10^{-16}$
Log_Income	1	2.1787	101.50	$<10^{-16}$
Residuals	317	6.8048		

Final Model

The estimated regression function for low-density states is:

$$\hat{Y}_i = -48.39 + 0.0158X_1 + 0.00916X_2 + 0.124X_3 - 0.380X_4 + 0.0846X_5 + 0.447X_6$$

where,

$\hat{Y}_i$ is the logarithm of rent

X_1 is the year

X_2 is the median building age

X_3 is the logarithm of unemployment rate

X_4 is the logarithm of vacancy rate

X_5 is the logarithm of square vacancy rate

X_6 is the logarithm of income

The estimated regression function for high-density states is:

$$\hat{Y}_i = -28.51 - 0.00546X_1 + 0.00755X_2 + 0.154X_3 - 0.193X_4 + 0.222X_5 + 0.718X_6$$

where,

$\hat{Y}_i$ is the logarithm of rent

X_1 is the year

X_2 is the median building age

X_3 is the logarithm of unemployment rate

X_4 is the logarithm of vacancy rate

X_5 is the logarithm of population density

X_6 is the logarithm of income

The effects of the variables and their chance of occurring under the null hypothesis can be evaluated with a standalone ANOVA. All included variables are significant at the 95% confidence level with F values that suggest a strong explanatory power.

Table 8: Low-Density Model ANOVA

	Df	Sum Sq	F Value	Pr(>F)
Year	1	0.9735	284.93	$<10^{-16}$
StructureAge	1	0.7766	227.32	$<10^{-16}$
Log_Employment	1	0.4425	129.51	$<10^{-16}$
Log_Vacancy	1	0.0420	12.28	0.00053
Log_Vacancy_sq	1	0.0367	10.73	0.00118
Log_Income	1	0.8412	246.22	$<10^{-16}$
Residuals	301	1.0284		

Table 9: High-Density Model ANOVA

	Df	Sum Sq	F Value	Pr(>F)
Year	1	0.0860	14.26	0.00019
StructureAge	1	0.9019	149.50	$<10^{-16}$
Log_Employment	1	0.6058	100.42	$<10^{-16}$
Log_Vacancy	1	0.6370	105.59	$<10^{-16}$
Log_Density	1	4.5776	758.81	$<10^{-16}$
Log_Income	1	2.1787	361.16	$<10^{-16}$
Residuals	315	1.9003		

Discussion & Conclusions

Interpretation of results

Our analysis revealed distinct differences in the influence of various predictor variables on rental prices, with Log_Density showing the most significant difference between models. In the low-density model, Log_Density was not a significant predictor, as evidenced by the near-0 trend left in the Component-Residual plot and the lack of explanatory power during model selection. Conversely, Log_Density shows a profound impact in the high-density model, with an F-value of 758.81 and a p-value of less than 10^{-16}. This stark contrast highlights the difference in influence of population density on rental prices between low-density and high-density states.

Implications of findings

The findings here demonstrate a clear trend: population density significantly influences rental prices in high-density states. This outcome aligns with the hypothesis that competition for housing in densely populated areas raises rental prices. Future research can verify the causal relationships behind this trend, potentially validating, in high-density states, the popular claim that increasing housing supply can mitigate rental prices.

Limitations of the study

The model failed to capture rental price trends in three states, indicating that the variables included in our dataset do not fully represent the complexities of the U.S. rental market. This limitation suggests the need for incorporating additional variables in future studies to better understand the dynamics of the rental market. These variables could include factors like the proportion of rental housing compared to overall housing, which is theorized to be a large impactor for the excluded states.

References

Vermont Housing Finance Agency. (2008). Housing Wages 2008. Retrieved from https://www.vhfa.org/documents/housing-wages-2008.pdf

Nau, R. (n.d.). The Log Transformation in Regression and ANOVA. Duke University. Retrieved from https://people.duke.edu/~rnau/411log.htm

Appendix

Appendix A: Data Preprocessing

The necessity of log-transformations was done using descriptive statistics and a comparison of the four possible transformations of each variable. This helped justify the transformation of the independent variables as well as the dependent variable. The log-log transformation was the best fit for all four scrutinized variables, having the lowest BIC and AIC for each.

Table A1: Log-Transformations Analysis

Density

	df	BIC	AIC
Log vs. Log	3	-353	-367
Log Rent vs. Normal	3	-537	-551
Normal vs. Log Density	3	10055	10041
Normal vs. Normal	3	10055	10041

Year

	df	BIC	AIC
Log vs. Log	3	-193	-207
Log Rent vs. Normal	3	-193	-207
Normal vs. Log Density	3	10522	10508
Normal vs. Normal	3	10522	10508

Vacancy

	df	BIC	AIC
Log vs. Log	3	-252	-266
Log Rent vs. Normal	3	-210	-225
Normal vs. Log Density	3	10435	10421
Normal vs. Normal	3	10480	10466

Income

	df	BIC	AIC
Log vs. Log	3	-503	-517
Log Rent vs. Normal	3	-538	-553
Normal vs. Log Density	3	10188	10174
Normal vs. Normal	3	10142	10128

Employment and Vacancy show a moderate reduction in skew, from moderate to weakly skewed. Density shows the largest reduction in skew, going from 2.24 to -0.14. Rent is reduced from 1.28 to 0.69, also a strong reduction in skew. Income goes from 0.47 to 0.07 in skew.

These are strong reductions in skewness, and help justify the decision made to log-transform the predictor variables.

Table A2: Descriptive Statistics for Log-Transformations

	mean	median	sd	min	max	skewness
Year	2014	2014	4.90	2006	2022	0.00
StructureAge	1976	1977	8.57	1954	1997	-0.38
Log_Employment	1.65	1.63	0.38	0.73	2.62	0.10
Log_Density	4.59	4.63	1.26	2	7	-0.14
Log_Vacancy	2.03	2.05	0.38	0.88	2.90	-0.26
Log_Income	11.13	11	0.17	11	12	0.07
Log_Rent	6.43	6.38	0.24	5.96	7.19	0.69

Appendix B: Automatic Model Selection

In addition to AIC, Mallow's C_p was employed to capture other potential leading models that may have been overlooked. The full models were chosen, consistent with the AIC results. The results of the C_p analysis show that Log_Density is the least relevant to the low-density model, with the explanatory power only barely offsetting the increase in complexity. There is a significant increase in C_p from the first high-density model choice to the second, going from 7.0 to 19.3, showing that all included variables effectively explain the response variable.

Table B1: Low-Density C_p

Low-Density Included Variables							Size	Cp
Year	StructureAge	Log_Employment	Log_Income	Log_Density	Log_Vacancy	Log_Vacancy_Sq	7	8.00
Year	StructureAge	Log_Employment	Log_Income	Log_Vacancy	Log_Vacancy_Sq		6	8.36
Year	StructureAge	Log_Employment	Log_Income	Log_Vacancy			5	17.14
Year	StructureAge	Log_Employment	Log_Income	Log_Density	Log_Vacancy		6	17.78
Year	StructureAge	Log_Employment	Log_Income	Log_Vacancy_Sq			5	18.70

Table B2: High-Density Cp

High-Density Included Variables						Size	Cp
Log_Density	**Log_Vacancy**	**Log_Income**	**StructureAge**	**Log_Employment**	**Year**	**6**	**7.0**
Log_Density	Log_Vacancy	Log_Income	StructureAge	Log_Employment		5	19.3
Log_Density	Log_Vacancy	Log_Income	StructureAge	Year		5	105.4
Log_Density	Log_Vacancy	Log_Income	StructureAge			4	106.0
Log_Density	Log_Income	StructureAge	Log_Employment	Year		5	110.6

The Bayesian Information Criterion, similar to AIC but with a larger penalty term for our sample size, does not select the model with the most variables for either model. All top models have very similar BIC values, which may suggest that including fewer variables can still have strong explanatory power for Log_Rent.

Table B3: Low-Density BIC

Low-Density Included Variables							size	BIC
Log_Income	**Log_Vacancy**	**Log_Vacancy_sq**	**Log_Employment**	**Year**	**StructureAge**		**6**	**-836.35**
Log_Income	Log_Density	Log_Vacancy	Log_Vacancy_sq	Log_Employment	Year	StructureAge	7	-833.03
Log_Income	Log_Employment	Year	StructureAge				4	-831.75
Log_Income	Log_Vacancy	Log_Employment	Year	StructureAge			5	-831.29
Log_Income	Log_Vacancy_sq	Log_Employment	Year	StructureAge			5	-829.76

Table B4: High-Density BIC

High-Density Included Variables							size	BIC
Log_Income	**Log_Density**	**Log_Vacancy**	**Log_Employment**	**Year**	**StructureAge**		**6**	**-836.35**
Log_Income	Log_Density	Log_Vacancy	Log_Employment	StructureAge		StructureAge	7	-833.03
Log_Income	Log_Density	Log_Vacancy	StructureAge				4	-831.75
Log_Income	Log_Density	Log_Vacancy	Year	StructureAge			5	-831.29
Log_Income	Log_Density	Log_Employment	Year	StructureAge			5	-829.76

Appendix C: R Code

```
---
title: "Exploratory"
author: "Breck Emert"
date: "`r Sys.Date()`"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)

library(readxl)
library(ggplot2)
library(ggcorrplot)
library(GGally)
library(dplyr)
library(tools)
library(gridExtra)
library(grid)
library(moments)
library(purrr)
library(RColorBrewer)
library(SimDesign)
library(e1071)
library(patchwork)
library(car)

palette <- brewer.pal(9, name = "YlGnBu")
primColor <- palette[6]
darkPrimColor <- palette[7]
```
```

```
# Convert colors to RGB values
primColor_rgb <- col2rgb(primColor)
darkPrimColor_rgb <- col2rgb(darkPrimColor)

# Print RGB values
print(paste("primColor RGB:", paste(primColor_rgb[,1], collapse = ",")))
print(paste("darkPrimColor RGB:", paste(darkPrimColor_rgb[,1], collapse = ",")))
```

```{r load}
filepath <- "C:/Users/Breck/Documents/Homework/Stat 840 Linear
Algebra/Final/Data/Full_Dataset.xlsx"
df <- read_excel(filepath)

# Set income in thousands for clarity
df$Income <- df$Income / 1000
# Make Rent last for graphing purposes
df <- df[c(setdiff(names(df), "Rent"), "Rent")]

df_num <- df[sapply(df, is.numeric)]

head(df)

```

```{r summary, fig.height=7, fig.width=9}
# Function to calculate descriptive statistics
descriptive_stats <- function(x) {
  c(mean = mean(x, na.rm = TRUE),
    median = median(x, na.rm = TRUE),
    sd = sd(x, na.rm = TRUE),
```

```
    min = min(x, na.rm = TRUE),
    max = max(x, na.rm = TRUE),
    skewness = skewness(x, na.rm = TRUE))
}

# Apply descriptive statistics function
stats_table <- as.data.frame(lapply(df_num, descriptive_stats))
stats_table <- t(stats_table)
print(stats_table)



# Scatterplot Matrix and Correlation Matrix
ggpairs(df_num, progress=FALSE, title="Scatterplot and Correlation Matrices")



# Histograms
# Calculate number of bins
n <- nrow(df)
n <- ceiling(sqrt(n))

# Histogram function
create_histogram <- function(var_name) {
  ggplot(df, aes_string(x = var_name)) +
    geom_histogram(bins = n, fill = primColor, color = darkPrimColor) +
    theme_minimal()
}

# Apply histogram function
plot_list <- lapply(names(df_num)[-1], create_histogram)

# Arrange 3x2 layout of histograms
```

```
grid.arrange(
  grobs = plot_list,
  ncol = 3,
  top = textGrob("Histogram Variable Distributions", gp = gpar(fontface = "bold", fontsize = 20))
)
```

```{r scatters, fig.height=9, fig.width=12}
# Scatterplots
response <- "Rent"
predictors <- setdiff(names(df_num), response)

# Plotting function
plot_function <- function(predictor) {
  ggplot(df_num, aes_string(x = predictor, y = response)) +
  geom_point() +
    labs(title = paste(response, "vs", predictor),
         x = predictor,
         y = response)
}

# Create a list of plots
plots <- map(predictors, plot_function)
grid.arrange(
  top = textGrob("Scatterplot Matrix of Rent vs. Predictors", gp = gpar(fontface = "bold",
fontsize = 20)),
  grobs = plots,
  nrow = 3,
  ncol = 2
)
```

```
# Log scatterplots
logResponse <- log(df_num$Rent, base=20)
predictors <- setdiff(names(df_num), response)

# Plotting function
logplot_function <- function(predictor) {
  ggplot(df_num, aes_string(x = predictor, y = logResponse)) +
  geom_point() +
    labs(title = paste("Log Rent", "vs", predictor),
         x = predictor,
         y = "Log Rent")
}

# Create a list of plots
plots <- map(predictors, logplot_function)
grid.arrange(
  top = textGrob("Scatterplot Matrix of Log Rent vs. Predictors", gp = gpar(fontface = "bold",
fontsize = 20)),
  grobs = plots,
  nrow = 3,
  ncol = 2
)


# Log-log scatterplots
logResponse <- log(df_num$Rent, base=20)
predictors <- setdiff(names(df_num), "Rent")

# Plotting function
```

```
loglogplot_function <- function(predictor) {
  df_num[[paste("log", predictor, sep = "")]] <- log(df_num[[predictor]], base=20)
  ggplot(df_num, aes_string(x = paste("log", predictor, sep = ""), y = "logResponse")) +
  geom_point() +
    labs(title = paste("Log Rent", "vs Log", predictor),
         x = paste("Log", predictor, sep = " "),
         y = "Log Rent")
}

# Apply plotting function
plots <- map(predictors, loglogplot_function)
grid.arrange(
  top = textGrob("Scatterplot Matrix of Log Rent vs. Log Predictors", gp = gpar(fontface =
"bold", fontsize = 20)),
  grobs = plots,
  nrow = 3,
  ncol = 2
)

# StructureAge Scatterplots
response <- "StructureAge"
predictors <- setdiff(names(df_num), response)

# Plotting function
plot_function <- function(predictor) {
  ggplot(df_num, aes_string(x = predictor, y = response)) +
  geom_point() +
  labs(title = paste("StructureAge", "vs", predictor),
       x = predictor,
       y = "StructureAge")
}
```

```
# Apply plotting function
plots <- map(predictors, plot_function)
grid.arrange(
  grobs = plots,
  nrow = 3,
  ncol = 2
)
```

```{r logDensity}
# Log transformation
df$logRent <- log(df$Rent)
df$logDensity <- log(df$Density)

# Models
model_log_log <- lm(logRent ~ logDensity, data = df)
model_log_none <- lm(logRent ~ Density, data = df)
model_none_log <- lm(Rent ~ Density, data = df)
model_none_none <- lm(Rent ~ Density, data = df)

# Compare
AIC(model_log_log, model_log_none, model_none_log, model_none_none)
BIC(model_log_log, model_log_none, model_none_log, model_none_none)
```

```{r logYear}
# Log transformation
df$logRent <- log(df$Rent)
df$logYear <- log(df$Year)
```

```
# Compare
model_log_log_year <- lm(logRent ~ logYear, data = df)
model_log_none_year <- lm(logRent ~ Year, data = df)
model_none_log_year <- lm(Rent ~ logYear, data = df)
model_none_none_year <- lm(Rent ~ Year, data = df)

# Compare
AIC(model_log_log_year, model_log_none_year, model_none_log_year,
model_none_none_year)
BIC(model_log_log_year, model_log_none_year, model_none_log_year,
model_none_none_year)
```

```{r logVacancy}
# Log transformation for Vacancy
df$logVacancy <- log(df$Vacancy)

# Compare
model_log_log_vacancy <- lm(logRent ~ logVacancy, data = df)
model_log_none_vacancy <- lm(logRent ~ Vacancy, data = df)
model_none_log_vacancy <- lm(Rent ~ logVacancy, data = df)
model_none_none_vacancy <- lm(Rent ~ Vacancy, data = df)

# Compare
AIC(model_log_log_vacancy, model_log_none_vacancy, model_none_log_vacancy,
model_none_none_vacancy)
BIC(model_log_log_vacancy, model_log_none_vacancy, model_none_log_vacancy,
model_none_none_vacancy)
```

```{r logIncome}
# Log transformation for Income
```

```
df$logIncome <- log(df$Income)

# Compare
model_log_log_income <- lm(logRent ~ logIncome, data = df)
model_log_none_income <- lm(logRent ~ Income, data = df)
model_none_log_income <- lm(Rent ~ logIncome, data = df)
model_none_none_income <- lm(Rent ~ Income, data = df)

# Compare
AIC(model_log_log_income, model_log_none_income, model_none_log_income,
model_none_none_income)
BIC(model_log_log_income, model_log_none_income, model_none_log_income,
model_none_none_income)
```

```
---
title: "Exploratory"
author: "Breck Emert"
date: "`r Sys.Date()`"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)

library(readxl)
library(ggplot2)
library(ggcorrplot)
library(GGally)
library(dplyr)
library(tools)
```
```

```
library(gridExtra)
library(grid)
library(moments)
library(purrr)
library(RColorBrewer)
library(SimDesign)

palette <- brewer.pal(9, name = "YlGnBu")
primColor <- palette[6]
darkPrimColor <- palette[7]
```

```{r load}
filepath <- "C:/Users/Breck/Documents/Homework/Stat 840 Linear
Algebra/Final/Data/Full_Dataset.xlsx"
df <- read_excel(filepath)

# Make Rent last for graphing purposes
df <- df[c(setdiff(names(df), "Rent"), "Rent")]

# Log df
columns_to_log <- c("Density", "Vacancy", "Income", "Rent", "Employment")
logdf <- log(df[columns_to_log])
names(logdf) <- paste0("Log_", columns_to_log)
logdf <- cbind(df[setdiff(names(df), columns_to_log)], logdf)

# Select only numeric
#logdf_num <- logdf[sapply(logdf, is.numeric)]

# REMOVE 2020 ONWARDS
logdf_num <- subset(logdf, Year < 2020)
```

```
# REMOVE STATE
#logdf_num <- subset(logdf_num, State != "Mississippi")[-2]

head(logdf_num[order(logdf_num$Log_Income), ])
```

```{r summary, fig.height=7, fig.width=9}
# Function to calculate descriptive statistics
descriptive_stats <- function(x) {
  c(mean = mean(x, na.rm = TRUE),
    median = median(x, na.rm = TRUE),
    sd = sd(x, na.rm = TRUE),
    min = min(x, na.rm = TRUE),
    max = max(x, na.rm = TRUE),
    skewness = skewness(x, na.rm = TRUE))
}

# Apply descriptive statistics function
stats_table <- as.data.frame(lapply(logdf_num[-c(1, 2)], descriptive_stats))
stats_table <- t(stats_table)
print(stats_table)


# Scatterplot Matrix and Correlation Matrix
ggpairs(logdf_num[-c(1, 2)], progress=FALSE, title="Scatterplot and Correlation Matrices")


# Histograms
# Calculate number of bins
```

```
n <- nrow(df)
n <- ceiling(sqrt(n))

# Histogram function
create_histogram <- function(var_name) {
  ggplot(logdf_num, aes_string(x = var_name)) +
    geom_histogram(bins = n, fill = primColor, color = darkPrimColor) +
    theme_minimal()
}

# Apply histogram function
plot_list <- lapply(names(logdf_num)[-c(1, 2)], create_histogram)
grid.arrange(
  grobs = plot_list,
  ncol = 3,
  top = textGrob("Histogram Variable Distributions", gp = gpar(fontface = "bold", fontsize = 20))
)

```

```{r scatters, fig.height=9, fig.width=12}
# Scatterplots
response <- "Log_Rent"
predictors <- setdiff(names(logdf_num), response)

# Plotting function
plot_function <- function(predictor) {
  ggplot(logdf_num, aes_string(x = predictor, y = response)) +
  geom_point() +
    labs(title = paste(response, "vs", predictor),
         x = predictor,
```

```
        y = response)
}

# Apply plot function
plots <- map(predictors, plot_function)
grid.arrange(
  top = textGrob("Scatterplot Matrix of Rent vs. Predictors", gp = gpar(fontface = "bold",
fontsize = 20)),
  grobs = plots,
  ncol = 2
)

```

```
---
title: "Modeling"
author: "Breck Emert"
date: "`r Sys.Date()`"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)

library(dplyr)
library(readxl)
library(ggplot2)
library(ggpubr)
library(lubridate)
library(car)
```
```

```
library(gridExtra)
library(moments)
library(broom)
library(forecast)
library(leaps)
library(MASS)
library(ggcorrplot)
library(grid)
library(purrr)
library(SimDesign)
library(RColorBrewer)
library(knitr)
library(caret)
library(mlbench)
library(patchwork)
library(plotly)
library(GGally)
library(lmtest)
library(leaps)

```

```{r load}
filepath <- "C:/Users/Breck/Documents/Homework/Stat 840 Linear
Algebra/Final/Data/Full_Dataset.xlsx"
df <- read_excel(filepath)

# Make Rent last for graphing purposes
df <- df[c(setdiff(names(df), "Rent"), "Rent")]

# Log df
```

```
columns_to_log <- c("Employment", "Density", "Vacancy", "Income", "Rent")
logdf <- log(df[columns_to_log])
names(logdf) <- paste0("Log_", columns_to_log)
logdf <- cbind(df[setdiff(names(df), columns_to_log)], logdf)

# Select only numeric
#logdf_num <- logdf[sapply(logdf, is.numeric)]
logdf_num <- logdf[-2]

# Split into low and high density dataframes
median_density <- median(logdf_num$Log_Density, na.rm = TRUE)
cat("Median Density of Overall Dataset:", median_density)

logdf_low <- subset(logdf_num, Log_Density <= median_density)
logdf_high <- subset(logdf_num, Log_Density > median_density)

# Add quadratic to low
logdf_low$Log_Vacancy_sq <- logdf_low$Log_Vacancy^2

# REMOVE 2020 ONWARDS
logdf_low <- subset(logdf_low, Year < 2020)
logdf_high <- subset(logdf_high, Year < 2020)

# REMOVE STATES
logdf_low <- subset(logdf_low, State != "Maine")
logdf_high <- subset(logdf_high, State != "Maine")

logdf_low <- subset(logdf_low, State != "Louisiana")
logdf_high <- subset(logdf_high, State != "Louisiana")

logdf_low <- subset(logdf_low, State != "Vermont")
```

```
logdf_high <- subset(logdf_high, State != "Vermont")

# Preview
cat("Length of low-density dataframe:", nrow(logdf_low),
    "\nLength of high-density dataframe:", nrow(logdf_high))
head(logdf_low[order(-logdf_low$Year), ])
```

```{r correlation analysis, fig.height=5, fig.width=11}
corr_matrix_low <- cor(logdf_low[-c(1, 9)])
corr_low <- ggcorrplot(corr_matrix_low, type = "lower", hc.order = FALSE, outline.color =
"white",
       lab = TRUE,
       lab_size = 4,
       tl.cex = 10) +
  theme(legend.text = element_text(size = 13)) +
  ggtitle("Low-Density States")

corr_matrix_high <- cor(logdf_high[-1])
corr_high <- ggcorrplot(corr_matrix_high, type = "lower", hc.order = FALSE, outline.color =
"white",
       lab = TRUE,
       lab_size = 4,
       tl.cex = 10) +
  theme(legend.text = element_text(size = 13)) +
  ggtitle("High-Density States")

corr_low + corr_high + plot_annotation(title = "Correlation Matrices of Low- and High-Density
States",
```

theme = theme(plot.title = element_text(size = 20, hjust = 0.5, margin = margin(b = 25))))

# VIF
lm_low_full <- lm(Log_Rent ~ ., data=logdf_low[-1])
cat("\nVIF for low-density\n")
vif(lm_low_full)

lm_high_full <- lm(Log_Rent ~ ., data=logdf_high[-1])
cat("\nVIF for high-density\n")
vif(lm_high_full)

```

```{r scatterplot matrices, fig.height=10, fig.width=10}
# Scatterplot Matrix
ggpairs(logdf_low[-1], progress=FALSE, title="Low Density: Scatterplot Matrix")
ggpairs(logdf_high[-1], progress=FALSE, title="High Density: Scatterplot Matrix")

```

```{r basic models}
lm_basic_low <- lm(Log_Rent ~ ., data=subset(logdf_low, select = -c(Log_Vacancy_sq, State)))
summary(lm_basic_low)

lm_basic_low_sq <- lm(Log_Rent ~ ., data=subset(logdf_low, select = -c(Log_Density, State)))
summary(lm_basic_low_sq)

lm_basic_high <- lm(Log_Rent ~ ., data=logdf_high[-1])

summary(lm_basic_high)

# VIF
vif(lm_basic_low_sq)
vif(lm_basic_high)

```

```{r AIC low}
library(leaps)
library(knitr)
# Perform AIC

# Chosen predictors
chosen_predictors <- c("Log_Income", "Log_Density", "Log_Vacancy", "Log_Vacancy_sq",
"Log_Employment", "Year", "StructureAge")

# Perform regsubsets for the chosen predictors
formula <- as.formula(paste("Log_Rent ~", paste(chosen_predictors, collapse=" + ")))
models <- regsubsets(formula, data=logdf_low, nbest=7)
models_summary <- summary(models)

# Empty df for storage
model_info <- data.frame(size = integer(), AIC = double(), row.names = character())

# Calculate AIC for the models
for (i in 1:length(models_summary$cp)) {
  included_predictors <- chosen_predictors[models_summary$which[i,][-1]]
  formula <- as.formula(paste("Log_Rent ~", paste(included_predictors, collapse=" + ")))
  model <- lm(formula, data=logdf_low)
  aic_value <- AIC(model)

```
  # Append the model information
  model_info <- rbind(model_info, data.frame(size = sum(models_summary$which[i,][-1]), AIC
= aic_value, row.names = paste(included_predictors, collapse=" + ")))
}

# Order models by AIC and format
ordered_models <- model_info[order(model_info$AIC),]
ordered_models$AIC <- round(ordered_models$AIC, digits = 2)
kable(ordered_models, format = "html", table.attr = "style='width:70%;'", caption = "Low-
Density: AIC Model Selection")
```

```
```{r BIC low}
# Perform BIC

# Chosen predictors
chosen_predictors <- c("Log_Income", "Log_Density", "Log_Vacancy", "Log_Vacancy_sq",
"Log_Employment", "Year", "StructureAge")

# Perform regsubsets for the chosen predictors
formula <- as.formula(paste("Log_Rent ~", paste(chosen_predictors, collapse=" + ")))
models <- regsubsets(formula, data=logdf_low, nbest=7)
models_summary <- summary(models)

# Empty df for storage
model_info <- data.frame(size = integer(), BIC = double(), row.names = character())

# Calculate BIC for the models
for (i in 1:length(models_summary$cp)) {
```
```

```
  included_predictors <- chosen_predictors[models_summary$which[i,][-1]]
  formula <- as.formula(paste("Log_Rent ~", paste(included_predictors, collapse=" + ")))
  model <- lm(formula, data=logdf_low)
  BIC_value <- BIC(model)

  # Append the model information
  model_info <- rbind(model_info, data.frame(size = sum(models_summary$which[i,][-1]), BIC
= BIC_value, row.names = paste(included_predictors, collapse=" + ")))
}

# Order models by BIC and format
ordered_models <- model_info[order(model_info$BIC),]
ordered_models$BIC <- round(ordered_models$BIC, digits = 2)
kable(ordered_models, format = "html", table.attr = "style='width:70%;'", caption = "Low-
Density: BIC Model Selection")
```

```{r AIC high}
# Perform AIC

# Chosen predictors
chosen_predictors <- c("Log_Income", "Log_Density", "Log_Vacancy", "Log_Employment",
"Year", "StructureAge")

# Perform regsubsets for the chosen predictors
formula <- as.formula(paste("Log_Rent ~", paste(chosen_predictors, collapse=" + ")))
models <- regsubsets(formula, data=logdf_high, nbest=7)
models_summary <- summary(models)

# Empty df for storage
```

```
model_info <- data.frame(size = integer(), AIC = double(), row.names = character())

# Calculate AIC for the models
for (i in 1:length(models_summary$cp)) {
  included_predictors <- chosen_predictors[models_summary$which[i,][-1]]
  formula <- as.formula(paste("Log_Rent ~", paste(included_predictors, collapse=" + ")))
  model <- lm(formula, data=logdf_high)
  aic_value <- AIC(model)

  # Append the model information
  model_info <- rbind(model_info, data.frame(size = sum(models_summary$which[i,][-1]), AIC
= aic_value, row.names = paste(included_predictors, collapse=" + ")))
}

# Order models by AIC and format
ordered_models <- model_info[order(model_info$AIC),]
ordered_models$AIC <- round(ordered_models$AIC, digits = 2)
kable(ordered_models, format = "html", table.attr = "style='width:70%;'", caption = "Model
Selection based on AIC")
```

```{r BIC high}
# Perform BIC

# Chosen predictors
chosen_predictors <- c("Log_Income", "Log_Density", "Log_Vacancy", "Log_Employment",
"Year", "StructureAge")

# Perform regsubsets for the chosen predictors
formula <- as.formula(paste("Log_Rent ~", paste(chosen_predictors, collapse=" + ")))
```

```
models <- regsubsets(formula, data=logdf_high, nbest=7)
models_summary <- summary(models)

# Empty df for storage
model_info <- data.frame(size = integer(), BIC = double(), row.names = character())

# Calculate BIC for the models
for (i in 1:length(models_summary$cp)) {
  included_predictors <- chosen_predictors[models_summary$which[i,][-1]]
  formula <- as.formula(paste("Log_Rent ~", paste(included_predictors, collapse=" + ")))
  model <- lm(formula, data=logdf_high)
  BIC_value <- BIC(model)

  # Append the model information
  model_info <- rbind(model_info, data.frame(size = sum(models_summary$which[i,][-1]), BIC
= BIC_value, row.names = paste(included_predictors, collapse=" + ")))
}

# Order models by BIC and format
ordered_models <- model_info[order(model_info$BIC),]
ordered_models$BIC <- round(ordered_models$BIC, digits = 2)
kable(ordered_models, format = "html", table.attr = "style='width:70%;'", caption = "Model
Selection based on BIC")

```

```{r Cp low}
# Perform Cp

# Cp on all possible predictors
models <- regsubsets(Log_Rent ~ ., data=logdf_low[-1], nbest=7)
```

```
models_summary <- summary(models)

# Print model variables ordered by R^2
model_info <- data.frame(models_summary$which, Cp=models_summary$cp)
ordered_models <- model_info[order(model_info$Cp), -1]
print(ordered_models)



# Summarize the lowest cp model
cp_lm <- lm(Log_Rent ~ Year + StructureAge + Log_Employment + Log_Vacancy +
Log_Vacancy_sq + Log_Income + Log_Density, data=logdf_low[-1])
summary(cp_lm)

qqnorm(residuals(cp_lm))
qqline(residuals(cp_lm))



# Chosen final model, for now
final_model_low <- cp_lm

```

```{r Cp high}
# Perform Cp

# Cp on all possible predictors
models <- regsubsets(Log_Rent ~ ., data=logdf_high[-c(1, 9)], nbest=6)
models_summary <- summary(models)

# Print model variables ordered by R^2
model_info <- data.frame(models_summary$which, Cp=models_summary$cp)
```

```
ordered_models <- model_info[order(model_info$Cp), -1]
print(ordered_models)


# Summarize the lowest cp model
cp_lm <- lm(Log_Rent ~ Year + StructureAge + Log_Employment + Log_Vacancy +
Log_Density + Log_Income + Log_Density, data=logdf_high)
summary(cp_lm)

qqnorm(residuals(cp_lm))
qqline(residuals(cp_lm))


# Chosen final model, for now
final_model_high <- cp_lm
summary(final_model_high)
```

```{r Preemptive Prplots, fig.height=9, fig.width=7}
library(lmtest)
# Check linear assumption for sample models

final_model_low <- lm(Log_Rent ~ Year + StructureAge + Log_Employment + Log_Vacancy +
Log_Vacancy_sq + Log_Income, data=logdf_low)

shapiro.test(residuals(lm_basic_low))
shapiro.test(residuals(final_model_low))
shapiro.test(residuals(final_model_high))

quiet(crPlots(lm_basic_low))
```

```
quiet(crPlots(lm_basic_low, main="Low-Density Model: Cr Plots"))
quiet(crPlots(final_model_high, main="High-Density Model: Cr Plots"))

qqnorm(residuals(final_model_low), main="Low-Density Model: Q-Q Plot")
qqline(residuals(final_model_low))

qqnorm(residuals(final_model_high), main="High-Density Model: Q-Q Plot")
qqline(residuals(final_model_high))

plot(fitted(final_model_low), residuals(final_model_low), xlab = "Fitted Values", ylab =
"Residuals", main = "Low-Density Model: Residuals vs. Fitted Values")
abline(h = 0, col = "red")

plot(fitted(final_model_high), residuals(final_model_high), xlab = "Fitted Values", ylab =
"Residuals", main = "High-Density Model: Residuals vs. Fitted Values")
abline(h = 0, col = "red")

bptest(final_model_high, ~ fitted(final_model_high), data = logdf_high)
```

```{r outliers high, fig.height=9, fig.width=7}
# Add identification column
logdf_low$ID <- seq_along(logdf_low$Log_Rent)

# Label the points; optional
plot(fitted(lm_basic_high), residuals(lm_basic_high), xlab = "Fitted Values", ylab = "Residuals",
main = "High-Density: Residuals vs. Fitted Values")
abline(h = 0, col = "red")
with(logdf_high, text(fitted(lm_basic_high), residuals(lm_basic_high), labels = State, pos = 4,
cex = 0.7))
```

```
# Calculate Cook's distance for the model
cooks_distance <- cooks.distance(lm_basic_high)

# Plot Cook's distance
plot(cooks_distance, pch = "*", cex = 2, main = "Cook's distance")
abline(h = 4 / length(cooks_distance), col = "red")

# Indices of Cook's distances greater than 4/n
threshold <- 4 / length(cooks_distance)
high_cooks_indices <- which(cooks_distance > threshold)

print(logdf_high[high_cooks_indices, ])
```

```{r outliers low, fig.height=9, fig.width=7}
# Add identification column
logdf_low$ID <- seq_along(logdf_low$Log_Rent)

# Label the points; optional
plot(fitted(final_model_low), residuals(final_model_low), xlab = "Fitted Values", ylab =
"Residuals", main = "Low Density: Residuals vs. Fitted Values")
abline(h = 0, col = "red")
#with(logdf_low, text(fitted(final_model_low), residuals(final_model_low), labels = State, pos =
4, cex = 0.7))

# Calculate Cook's distance for the model
cooks_distance <- cooks.distance(final_model_low)

# Plot Cook's distance
plot(cooks_distance, pch = "*", cex = 2, main = "Cook's distance")
```

```
abline(h = 4 / length(cooks_distance), col = "red")

# Indices of Cook's distances greater than 4/n
threshold <- 4 / length(cooks_distance)
high_cooks_indices <- which(cooks_distance > threshold)

print(logdf_low[high_cooks_indices, ])
```

```{r outliers high, fig.height=9, fig.width=7}
# Add identification column
logdf_high$ID <- seq_along(logdf_high$Log_Rent)

# Label the points; optional
plot(fitted(final_model_high), residuals(final_model_high), xlab = "Fitted Values", ylab =
"Residuals", main = "Residuals vs. Fitted Values")
abline(h = 0, col = "red")
with(logdf_low, text(fitted(final_model_high), residuals(final_model_high), labels = State, pos =
4, cex = 0.7))

# Calculate Cook's distance for the model
cooks_distance <- cooks.distance(final_model_high)

# Plot Cook's distance
plot(cooks_distance, pch = "*", cex = 2, main = "Cook's distance")
abline(h = 4 / length(cooks_distance), col = "red")

# Indices of Cook's distances greater than 4/n
threshold <- 4 / length(cooks_distance)
high_cooks_indices <- which(cooks_distance > threshold)
```

```
print(logdf_low[high_cooks_indices, ])
```

```{r vermont}
# Subset for Vermont
vermont_df <- subset(df, State == "Utah")

# Subset for other states
other_states_df <- subset(df, State != "Utah")

# Calculate means for Vermont
vermont_means <- colMeans(vermont_df[, sapply(vermont_df, is.numeric)], na.rm = TRUE)

# Calculate means for other states
other_states_means <- colMeans(other_states_df[, sapply(other_states_df, is.numeric)], na.rm = TRUE)

# Combine into a data frame
comparison_table <- data.frame(Vermont = vermont_means, Other_States = other_states_means)

# Print the table
print(comparison_table)

```

```{r more outliers}
library(plotly)
# Assuming lm_basic_low_sq is your linear model
residuals_df <- data.frame(Residuals = residuals(lm_basic_low_sq),
                           Standardized = rstandard(lm_basic_low_sq),
                           CooksD = cooks.distance(lm_basic_low_sq))
```

```
# Add an identifier
residuals_df$ID <- row.names(residuals_df)

# Identify potential outliers
outliers <- residuals_df[abs(residuals_df$Standardized) > 2, ]

# Look at the observations with the largest Cook's distances
influential <- residuals_df[order(-residuals_df$CooksD), ]

# Print out the top potential outliers and influential observations
print(head(outliers))
print(head(influential))



# Plotly residuals plot
res <- residuals(lm_basic_low_sq)
theoretical_quantiles <- qqnorm(res, plot.it = FALSE)$x
sample_quantiles <- sort(res)
years <- logdf_low$Year[order(res)]

# Calculate the line of best fit for the QQ plot
fit <- lm(sample_quantiles ~ theoretical_quantiles)
intercept <- coef(fit)[1]
slope <- coef(fit)[2]

# Create the plotly
p <- plot_ly() %>%
  add_markers(x = theoretical_quantiles, y = sample_quantiles, text = years,
              hoverinfo = 'text+y+x', name = 'Residuals') %>%
  add_lines(x = theoretical_quantiles, y = intercept + slope * theoretical_quantiles,
```

```
          line = list(color = 'red'), name = 'Fit Line') %>%
  layout(title = 'Normal Q-Q Plot',
         xaxis = list(title = 'Theoretical Quantiles'),
         yaxis = list(title = 'Sample Quantiles'))

p

```

```{r Cross Validation Low}
library(caret)
# Perform LOOCV

# Run with LOOCV control
control <- trainControl(method = "LOOCV")
model_loocv <- train(Log_Rent ~ StructureAge + Log_Employment + Log_Density +
Log_Vacancy + Log_Income + Year + Log_Vacancy_sq,
                     data = logdf_low,
                     method = "lm",
                     trControl = control)

# Results
print(model_loocv)



# Define 10-fold cross-validation
train_control <- trainControl(method = "cv", number = 10)

# Apply the train function with the defined train control
model_cv <- train(
```

```
  form = Log_Rent ~ StructureAge + Log_Employment + Log_Density + Log_Vacancy +
Log_Income + Year + Log_Vacancy_sq,
  data = logdf_low,
  method = "lm",
  trControl = train_control
)

print(model_cv)
```

```{r Cross Validation High}
# Perform LOOCV

# Run with LOOCV control
control <- trainControl(method = "LOOCV")
model_loocv <- train(Log_Rent ~ StructureAge + Log_Employment + Log_Density +
Log_Vacancy + Log_Income + Year,
                     data = logdf_high,
                     method = "lm",
                     trControl = control)

print(model_loocv)



# Define 10-fold cross-validation
train_control <- trainControl(method = "cv", number = 10)

# Apply the train function with the defined train control
model_cv <- train(
  form = Log_Rent ~ StructureAge + Log_Employment + Log_Density + Log_Vacancy +
Log_Income + Year,
```

```
  data = logdf_high,
  method = "lm",
  trControl = train_control
)

print(model_cv)
```

```{r ANOVA low}
Anova(final_model_low)

lm_basic_low <- lm(Log_Rent ~ Year + StructureAge + Log_Employment + Log_Vacancy +
Log_Vacancy_sq + Log_Income, data=logdf_low)
Anova(final_model_low, lm_basic_low)

lm_basic_low <- lm(Log_Rent ~ Year + StructureAge + Log_Employment + Log_Vacancy +
Log_Income, data=logdf_low)
Anova(final_model_low, lm_basic_low)

final_model_low <- lm(Log_Rent ~ Year + StructureAge + Log_Employment + Log_Vacancy +
Log_Vacancy_sq + Log_Income, data=logdf_low)
summary(final_model_low)

```

```{r ANOVA high}
Anova(final_model_high)

lm_basic_high <- lm(Log_Rent ~ StructureAge + Log_Employment + Log_Vacancy +
Log_Income, data=logdf_high)
Anova(final_model_high, lm_basic_high)
```

summary(final_model_high)
```
```