To what extent do violent crime rates impact house prices in Pennsylvania, USA in 2016?

Economics of Crime: BEE3074

700037998

Contents

1 Introduction

The hovering presence of violent crime rates over property markets raises pivotal questions about their impact on house prices in Pennsylvania's real estate market. Prospective home buyers in such environments carefully consider various factors influencing property prices, including violent crime rates (VCRs). Understanding this relationship is crucial for gaining insights into housing market dynamics, as crime rates directly influence buyers' perceptions of safety and property desirability, thereby affecting property values and market trends. Additionally, elevated crime rates can significantly impact community well-being and overall quality of life.

This essay seeks to address the multitude of inconclusive findings in the existing literature by empirically investigating the impact of violent crime rates on house prices, aiming to provide new insights into this relationship. Through analysis of Pennsylvania house price data and development of a machine learning model incorporating crime rates and relevant factors, this essay aims to offer valuable insights. Additionally, this research will discuss data limitations and methodological considerations while exploring potential policy implications. The objective is to shed light on the relationship between violent crime rates and house prices in Pennsylvania communities, proposing empirically supported strategies to address violent crime and inform evidence-based policy-making aimed at enhancing community safety and promoting housing market stability.

2 Background

In 2016, Pennsylvania reported a violent crime rate of 5.2 incidents per 100,000 people, showing a 1.35% increase from the previous year and surpassing the national average (The Pennsylvania Statistical Analysis Center, 2019). This upward trend in violent crime rates since 2014 (Berk, 2022) underscores the importance of closely examining its implications.

Crime rates have significant implications for housing markets, extending beyond instilling fear and insecurity among potential home buyers (Buonanno, Montolio, & Raya-Vílchez, 2012). Fear of crime, influenced by various neighborhood dynamics such as disinvestment, demolition, and deindustrialization, can accelerate neighborhood decline by weakening social controls and impacting urban development outcomes (Skogan, 1986). Areas with higher violent crime rates often experience decreased property values, limited investment in infrastructure, and challenges in attracting businesses and residents, all of which can influence housing prices (Hanson, Sawyer, Begle, & Hubel, 2010). Understanding this relationship is crucial for policy makers and informs the development of effective strategies that promote community development, equitable housing policies, crime prevention, urban planning, and investment attraction (Pope & Pope, 2011). Ensuring that research findings on the impact of violent crime on house prices translate into actionable policies is essential for improving societal outcomes.

3 Data

3.1 Data Cleaning

The dataset used in this essay, titled 'Philadelphia Real Estate' (PRE), was sourced from the Kaggle data repository ("Philadelphia Real Estate," 2017). Initial data cleaning involved removing duplicate columns and checking variables for missing values, duplicates, and correcting data type classifications. Additionally, variable names were standardized to comply with a uniform format.

Variables in the dataset with more than 10% zero values were excluded. Table 1 shows three variables exceeding this threshold and subsequently excluded. This method enhances data reliability and accuracy by reducing noise, thereby improving modeling effectiveness.

Name of Variable	Percentage of 0
Other	100%
Record Deed	91%
PGW	31%
Water	6%
year Built	2%

Table 1: Percentage of 0 Values in Variables with 0s

Additionally, all rows containing zero values were removed from the dataset, resulting in the exclusion of a relatively small number of rows (20), many of which contained multiple zero values. This step was critical for preserving data integrity and mitigating potential inaccuracies associated with missing or undefined data points.(See Appendix 1 for data cleaning details)

3.2 Data Overview

In the cleaned PRE dataset, there are 12 predictor variables and 523 observations. The dependent variable 'House price' in the dataset is measured in thousands of dollars ($1000), where each unit of 'House price' represents one thousand dollars. As shown in Table 2, the variables are classified into three data types, which is essential for ensuring accuracy and facilitating interpretation based on the nature of the data.

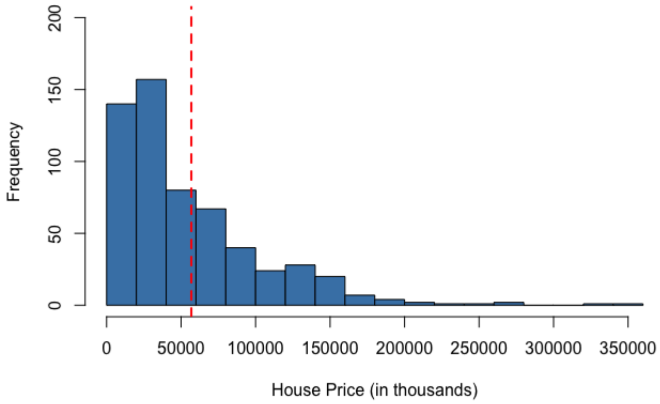
Data Classifications	Features
Categorical Data	Postcode, Year the House was Build
Discrete Data	Number of Bedrooms, Number of Bathrooms
Continuous Data	Sheriff Cost, Advertising, Water, Walking Score, Violent crime rate, School Score, Tax Assessment

Table 2: Data Type of Variables

3.3 Data Exploration

Figure 1 illustrates a range of house prices from $5,200 to $3,500,000. Upon closer examination, key measures of central tendency reveals insight into the distribution of these prices. The median ($40,000) and mode ($40,000) are notably situated to the left of the mean ($57,772), indicated by the red dashed line. This highlights the asymmetry of the house price distribution, with a significant concentration of prices clustering towards the lower end, contributing to the observed leftward skewness in the dataset.

Figure 1: **Distribution of House Prices**



Potential outliers have been detected in the response variable. Outlier tests revealed that approximately 2.9% of the observations in this variable exhibit values that deviate significantly from the majority of the data. Further examination of Figure 1, illustrating the distribution of house prices, along with the calculation of central tendencies, supports the decision to retain these outliers in the dataset.

Variable	Correlation with House Price
Sheriff Cost	0.98
Tax Assessment	0.81
School Score	0.49
Violent Crime Rate	-0.42
Walking and Transit Score	-0.40
Bathrooms	0.39
Square Feet	0.36
Year Built	0.35
Postcode	-0.11
Advertising	-0.10
Bedrooms	0.07
Water	0.02

Table 3: Correlation of Variables with House Price

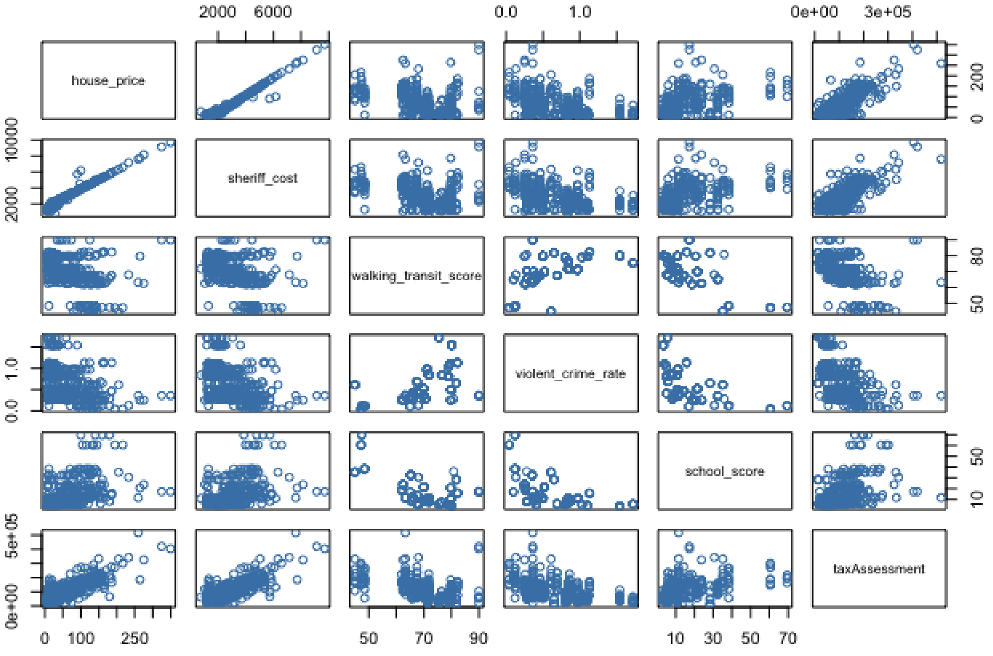
Table 5 presents correlation coefficients between the response variable 'House price' and all explanatory variables, highlighting notably strong correlations with 'sheriff cost' and 'tax assessment'. Concerned about these high correlations, an investigation into potential multicollinearity involved examining the top 5 correlated variables with each other and with the response variable.

To assess multicollinearity rigorously, variance inflation factor (VIF) scores were computed and summarized in Table 6. Importantly, all VIF scores are below 5, indicating no significant multicollinearity issues between the explanatory variables and the response variable. This analysis confirms the reliability of the regression model, affirming that the explanatory variables are sufficiently independent in explaining the variation in house prices.

Variable	Variance Inflation Factor
Tax Assessment	4.23
Sheriff Cost	2.89
School Score	2.65
Walking and Transit Score	2.23
Violent Crime Rate	1.68

Table 4: Variance Inflation Factors (VIF) for Predictor Variables in Relation to House Price

Figure 2: **Pairwise plot of of Key Variables**



The pairwise relationships among key variables, notable patterns emerge, as demonstrated in figure 2. Sheriff Cost shows a strong positive correlation with House Price, suggesting higher house prices in areas with elevated sheriff costs. Conversely, lower house prices are associated with longer commute distances. Additionally, higher house prices correlate with lower Violent Crime Rates, indicating safer neighborhoods in more affluent areas. School Scores exhibit consistency across varying house prices, implying a limited impact on housing market dynamics. Lastly, Tax Assessment demonstrates a weak positive correlation with House Price. These insights highlight the relationships shaping house prices within the dataset.

4 Method

4.1 Random Forest Models

4.1.1 Model Selection

This essay will utilize random forest regression due to its suitability to this dataset. This approach is robust against outliers and effectively handles datasets of approximately 500 observations. The model can accommodate a diverse mixture of variable types, aligning well with the varied data classifications indicated in the data section. It is well-suited to this dataset with diverse correlations, as it can capture complex relationships and identifies important features contributing to house prices. Additionally, the combination of decision trees in random forest improves generalization performance, enhancing the reliability of this model type and reducing risk of overfitting.

To train and evaluate the model effectively,the dataset is split into training and testing sets using an 80:20 ratio. This ensures a robust assessment of model performance on unseen data.

4.1.2 Model Refinement

Figure 1 shows that using a random forest model with 10 predictors strikes an optimal balance between complexity and performance, enhancing the model's effectiveness and reliability. In Figure 2, water and bedrooms, with correlations of less than 0.1 with house price, as revealed in the data section, are identified as the least important variables. Therefore, these variables will be excluded to streamline model complexity.

Figure 3: Optimal Predictor Count and Test MSE Relationship

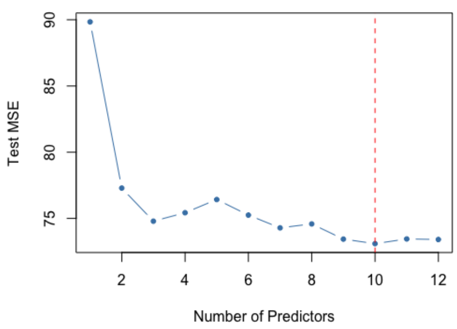
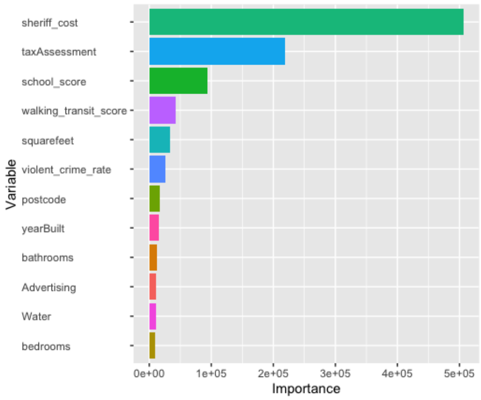


Figure 4: Variable Importance in Random Forest



Two random forest models were constructed to examine the impact of VCRs on house prices—one including this predictor and another excluding it. This approach directly assesses the influence of VCRs on house prices and enables a comparative analysis to quantify their specific contribution.

- Random Forest(Including Violent Crime Rate)

$$\hat{Y}_1 = f(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10})$$

- Random Forest(Excluding Violent Crime Rate)

$$\hat{Y}_2 = f(X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10})$$

where:

$$\begin{aligned}
\hat{Y}_1 &= \text{predicted house prices for model with violent crime rate} \\
\hat{Y}_2 &= \text{predicted house prices for model without violent crime rate} \\
f &= \text{random forest prediction function} \\
X_1 &= \text{violent\_crime\_rate} \\
X_2 &= \text{walk\_transit\_score} \\
X_3 &= \text{school\_score} \\
X_4 &= \text{taxAssessment} \\
X_5 &= \text{sheriff\_cost} \\
X_6 &= \text{Advertising} \\
X_7 &= \text{yearBuilt} \\
X_8 &= \text{squarefeet} \\
X_9 &= \text{bathrooms} \\
X_{10} &= \text{postcode}
\end{aligned}$$

The optimal number of K-fold cross-validation for two random forest models—one including VCRs and one excluding this predictor. The VCR-inclusive model performed best with 5 K-folds, while the VCR-exclusive model achieved optimal performance with 10 K-folds. This variation emphasizes the importance of the VCR predictor for enhancing model accuracy and highlights the need for tailored cross-validation strategies to optimize predictive performance based on model configuration.

The optimal tree depth differs between the models, with the crime-inclusive model achieving its best performance at a depth of 4, whereas the crime-exclusive model performs optimally with a depth of 1. The difference in optimal tree depth highlights how including the violent crime rate increases model complexity, with the crime-inclusive model benefiting from deeper trees to capture complex patterns in the data.

The number of trees in the crime-inclusive model (10,000) versus the MSE in the crime-exclusive model (1,000) highlighting the necessity of distinguishing how the inclusion of the violent crime rate variable distinctly shapes model performance under varying conditions, further highlighting the variable's importance in predictive accuracy.

Crime Inclusive MSE achieves its lowest value at MTRY = 10 (26.50), while Crime Exclusive MSE reaches its minimum at MTRY = 9 (148.46). Therefore, optimizing MTRY at both models maximum MTRY setting achieves better generalisations unseen data, particularly when considering the variable inclusion model.

These random forest regressions provide robust frameworks to examine how including the violent crime rate impacts house price predictions. These findings highlight the importance of this variable in enhancing model performance and accuracy, prompting further exploration through hypothesis testing and analysis. Detailed statistical tables corresponding to these analyses can be found in Appendix 2.

5 Results

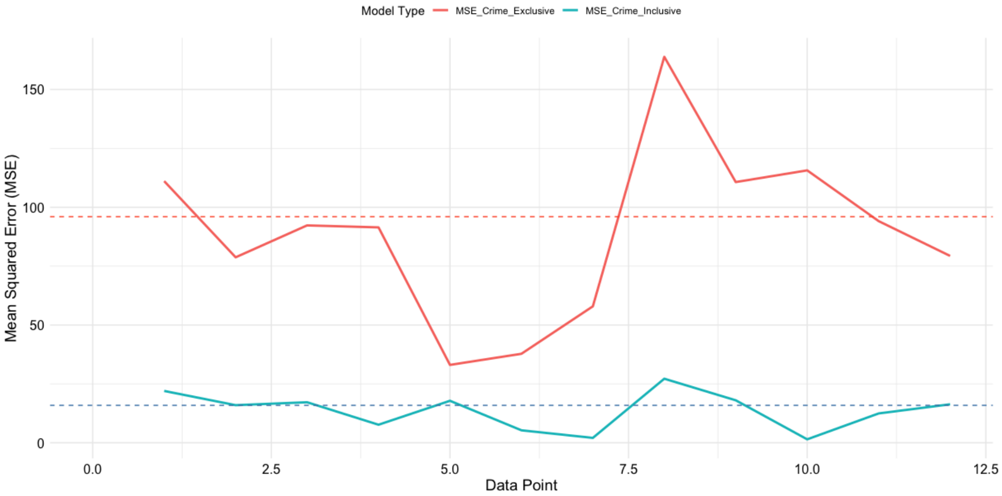
5.1 Model performance Metrics

Model Metric	RF crime inclusive	RF crime exclusive
MSE	15.90	96.00
RMSE	3.99	9.80
MAE	1.48	3.06
R-Squared	0.97	0.82

Table 5: Model Performance Metric Results

Table 6 highlights the superior predictive performance of the crime-inclusive model (MSE = 15.90) compared to the crime-exclusive model (MSE = 96.00), highlighting the importance of incorporating crime rates as a predictor in understanding neighborhood dynamics and housing market behavior. The lower MSE of the crime-inclusive model reflects its enhanced accuracy in capturing market complexities and buyer preferences in real estate predictions. Additionally, Figure 5 depicts the mean MSE values for both models, with dashed lines at MSE = 15.90 (crime-inclusive) and MSE = 96.00 (crime-exclusive), visually emphasizing the performance difference. This graphical representation reinforces the critical role of crime rates in housing market analysis, aligning with the findings from Table 6.

Figure 5: **MSE Comparison: Crime-Inclusive vs. Crime-Exclusive Models**



The RMSE of approximately $3,990 for the random forest model including crime suggests that, on average, the predictions deviate by this amount in terms of house prices. While this score indicates better accuracy compared to the model excluding crime, neither achieves exceptional precision. A low RMSE relative to house prices signifies prediction errors, suggesting underlying factors not fully captured. This complexity highlights the challenge of accurately predicting house prices.

The crime inclusive model has a lower MAE of 1.48, indicating that its predictions are closer to actual house prices compared to the crime exclusive model, which has an MAE of 3.06. Showing that the crime inclusive RF has better accuracy and prediction error magnitude than the other model, highlighting it has better predictive performance in capturing house price variability.

The crime-inclusive random forest model achieves an impressive R-squared value of 97%, demonstrating strong predictive accuracy by explaining a significant portion of the variance in predicted house prices.This indicates a robust relationship between predictors, including crime rates, and house prices, with the model showing a high fit to the training data.In contrast, the crime-exclusive model achieves an R-squared of 82%, indicating reduced explanatory power compared to the crime-inclusive model. This diminished performance highlights the crucial role of crime rates in predicting house prices and emphasizes that their exclusion results in a less accurate model.

5.2 Hypothesis Test

To evaluate the impact of violent crime rates on house prices in Pennsylvania, a comparative analysis using MSE values obtained from two models, crime inclusive and crime-exclusive. This approach allowed assessment of how the inclusion of violent crime rates influences the accuracy of our house price prediction model.

The hypothesis framework was established with the following hypothesis:

- Null Hypothesis:
$$H_0 : \text{MSE}_{\text{with}} = \text{MSE}_{\text{without}}$$
  - MSE_{with}: MSE for random forest inclusive of violent crime rates.
  - $\text{MSE}_{\text{without}}$: MSE for random forest exlusive of violent crime rates.
- Alternative Hypothesis:
$$H_1 : \text{MSE}_{\text{with}} \neq \text{MSE}_{\text{without}}$$

A paired t-test was performed to compare the distribution of MSE values between RF1 and RF2. The resulting p-value of 0.04635 indicates a statistically significant difference in mean MSE values between the crime-inclusive and crime-exclusive models. With a significance level alpha set at 0.05, the observed p-value falls below this threshold, leading to the rejection of the null hypothesis.

Interpreting the results within the context of our hypothesis framework:

- Given:
$$\alpha = 0.05$$
$$p = 0.04635$$
- Statistical Decision:
$$0.04635 < 0.05$$

This rejection suggests that including violent crime rates as a predictor significantly influences the predictive accuracy of our random forest model for house price estimations. The acceptance of the alternative hypothesis highlights the importance of incorporating crime data into predictive modeling to better understand neighborhood dynamics and housing market behavior.

6 Discussion

These results highlight the substantial impact of violent crime rates on house prices in Pennsylvania. The comparison of MSE values from crime-inclusive and crime-exclusive random forest models provides compelling evidence supporting the inclusion of violent crime rates as a predictor in house price prediction models.

The crime-inclusive model exhibited a significantly lower MSE of 15.90 compared to 96.00 for the crime-exclusive model, demonstrating the importance of incorporating crime rates to enhance predictive accuracy. Additionally, the crime-inclusive model showed improvements in other performance metrics, including a lower RMSE of approximately $3,990 and a substantially lower MAE of 1.48 compared to the crime-exclusive model.

Our hypothesis testing using a paired t-test further confirmed these findings, with a statistically significant difference in mean MSE values between the crime-inclusive and crime-exclusive models (p = 0.04635). This highlights the influence of violent crime rates on house price predictions and emphasizes the critical role of crime data in understanding housing market dynamics.

Therefore, these results highlight that incorporating violent crime rates significantly improves the accuracy and predictive performance of house price prediction models in Pennsylvania. By integrating crime data into housing market analysis, policymakers can make informed decisions aimed at enhancing community safety and promoting housing market stability. This approach facilitates the development of effective strategies to address crime, attract investments, and improve overall neighborhood well-being. Moreover, incorporating crime data in policy-making ensures that interventions are targeted and evidence-based, contributing to the creation of more resilient and equitable communities.

7 Limitations

The dataset's relatively small size may limit the generalizability of findings, while a larger dataset could yield more robust results. Outliers in the dataset, though manageable by the selected model, might have distorted analyses. In future studies with larger datasets, outliers could be identified and removed to ensure a more accurate representation of the relationship between violent crime rates and house prices.

Omitted variable bias may have occurred, influencing the relationship between violent crime rates and house prices are not included in the analysis.The omission of relevant socioeconomic or demographic factors could impact the predictive models outputs.

Furthermore, the need for additional variables is highlighted by the study's focus on model refinement and complexity. The inclusion of more informative variables could enhance the predictive capabilities of the models, particularly in understanding the complex relationship between violent crime rates and house prices. Also, with the model using maximum number of MTRY may not be optimal, so therefore the inclusion of more informative predictors would aid in model performance.

The limitations of the models highlight the importance of data quality and model refinement. Larger, diverse dataset, removal of outliers and consideration of additional variables, would enhance accuracy and insights into the impact of violent crime rates on house prices, improving the reliability and applicability of predictive models.

8 Policy implications

Implementing a policy that significantly increases funding for early childhood education programs in Pennsylvania would be a pivotal strategy in reducing VCRs and positively impact house prices. Early childhood education interventions have consistently demonstrated significant effects on cognitive development and social behaviors, equipping young children with foundational skills crucial for future success. Through this intervention, policymakers can effectively target the root causes of violent-crime by nurturing positive social interactions, emotional self-regulation, and conflict resolution skills early in life.

Research findings consistently highlight a strong correlation between participation in early education programs and reduced involvement in delinquent behaviors, later in life (Anwar & Derin, 2019; Yoshikawa, 1995). Children who benefit from quality education early are less likely to engage in violent activities as they mature, contributing to enhanced public safety (World Health Organization, 2009). Also, early education interventions, break cycles of inter generational poverty through offering vulnerable populations a pathway to economic stability and improved social outcomes (Harper, Marcus, & Moore, 2003).

Increased investment in early childhood education could positively influence neighborhood safety perceptions and property values in Pennsylvania. Safer communities, characterized by lower violent crime rates, are inherently more attractive to prospective home buyers and encourage investments in housing infrastructure (Tita, Petras, & Greenbaum, 2006). By addressing the underlying social determinants of crime through education, policymakers can create a more conducive environment for community development.

This policy approach highlights the inter-connectedness of education policy, crime prevention, and housing market resilience, emphasizing the importance of evidence-based policy making to drive positive economic and social outcomes. Through utilizing early childhood education as a preventive strategy, Pennsylvania can effectively tackle crime-related challenges and promote inclusive economic growth and community well-being.

9 Conclusion

In summary, this essay emphasizes the significant impact of violent crime rates on house prices in Pennsylvania during 2016. Elevated crime rates correlate with lower property values, influencing buyer perceptions of safety and neighborhood desirability. This highlights the importance of incorporating crime data into predictive models for informed policy-making aimed at enhancing community safety, such as interventions like increased investment in early childhood education to address underlying social determinants of crime.

10 Bibligoraphy

1. Anwar, A., & Derin, T. (2019). Early Childhood Education and Its Correlation with Crime: A Review. *Utamax : Journal of Ultimate Research and Trends in Education, 1*(1), 13–17. https://doi.org/10.31849/utamax.v1i1.2758

2. Berk, R. (2022). Is Violent Crime Increasing? — Department of Criminology. Retrieved March 23, 2024, from crim.sas.upenn.edu website: https://crim.sas.upenn.edu/fact-check/violent-crime-increasing

3. Buonanno, P., Montolio, D., & Raya-Vílchez, J. M. (2012). Housing prices and crime perception. *Empirical Economics, 45*(1), 305–321. https://doi.org/10.1007/s00181-012-0624-y

4. Hanson, R. F., Sawyer, G. K., Begle, A. M., & Hubel, G. S. (2010). The impact of crime victimization on quality of life. *Journal of Traumatic Stress, 23*(2), 189–197. https://doi.org/10.1002/jts.20508

5. Harper, C., Marcus, R., & Moore, K. (2003). Enduring Poverty and the Conditions of Childhood: Lifecourse and Intergenerational Poverty Transmissions. *Pergamon, 31*(3), 535–554. science direct. https://doi.org/0305-750

6. Philadelphia Real Estate. (2017, April). Retrieved April 12, 2024, from www.kaggle.com website: https://www.kaggle.com/datasets/harry007/philly-real-estate-data-set-sample

7. Pope, D., & Pope, J. (2011). Crime and property values: Evidence from the 1990s crime drop. *Elsevier, 2*(2), 185–197.

8. Skogan, W. (1986). Fear of Crime and Neighborhood Change. *Crime and Justice, 8*, 203–229. https://doi.org/10.1086/449123

9. The Pennsylvania Statistical Analysis Center. (2019). Pennsylvania a report of crime trends and statistics 2012-2016 (pp. 1–6). Retrieved from

10. Tita, G. E., Petras, T. L., & Greenbaum, R. T. (2006). Crime and Residential Choice: A Neighborhood Level Analysis of the Impact of Crime on Housing Prices. *Journal of Quantitative Criminology, 22*(4), 299–317. https://doi.org/10.1007/s10940-006-9013-z

11. World Health Organization. (2009). Preventing violence by developing life skills in children and adolescents Series of briefings on violence prevention. In *World Health Organization*, (pp. 3–6). Retrieved from https://iris.who.int/bitstream/handle/10665/44089/9789241597838$_e$*ng.pdf*

11 Appendix 1

Following the initial loading of the dataset in R, a series of data cleaning steps were undertaken to prepare the ”Philadelphia Real Estate” dataset sourced from Kaggle (”Philadelphia Real Estate,” 2017) for subsequent analysis. This appendix outlines the specific procedures employed to clean and refine the dataset.

Duplicate columns were identified and removed to streamline the dataset and eliminate redundancy. This step ensures that each variable in the dataset is unique and contributes distinct information.

```r
REC_Df <- REC_Df[, !duplicated(names(REC_Df))]
```

Null values and duplicate rows were assessed to ensure data completeness and consistency.

```r
# Check for null values
if (any(is.na(REC_Df))) {
  # Handle null values (if any)
}

# Check for duplicate rows
if (any(duplicated(REC_Df))) {
  # Remove duplicate rows
  REC_Df <- unique(REC_Df)
}
```

Erroneous zero values were identified and addressed by excluding columns with excessive zero values and removing rows containing zero values.

```r
# Exclude columns with >10% zero values
REC_Df <- REC_Df[, colMeans(REC_Df == 0) <= 0.1]

# Remove rows with zero values
REC_Df <- REC_Df[rowSums(REC_Df == 0) == 0, ]
```

Rows containing hyphens (indicating missing or incomplete data) were removed to enhance data integrity.Variable names were standardized to ensure uniformity and consistency across the dataset.Columns containing numerical data in character format were converted to numeric data types for analytical purposes.

```r
# Remove rows with hyphens
REC_Df <- REC_Df[!grepl("-", REC_Df$variable), ]
# Standardize variable names
names(REC_Df) <- make.names(names(REC_Df))
# Convert character columns to numeric
REC_Df$numeric_column <- as.numeric(as.character(REC_Df$numeric_column))
```

These data cleaning procedures were essential for preparing the ’Philadelphia Real Estate’ dataset for subsequent data analysis and modeling. By addressing issues such as duplicate columns, null values, erroneous zero values, and inconsistent data types, the cleaned dataset was made suitable for reliable and meaningful statistical analysis and interpretation.

12 Appendix 2

This table displays Mean Squared Error (MSE) values obtained by tuning the number of K-folds for both the crime-inclusive and crime-exclusive models. Notably, the crime-inclusive model achieved the lowest MSE of 37.07 with 10 K-folds and the crime exclusive model achieves lowest MSE with 5-Kfolds.

Number of K-folds	Crime Inclusive MSE	Crime Exclusive MSE
5	37.17	152.26
10	37.07	164.59
15	37.24	157.18

Table 6: Results of Tuning for K-folds

The crime-inclusive model performed optimally at a tree depth of 4, with a significantly lower MSE of 36.13 compared to other depths. In contrast, the crime-exclusive model achieved its lowest MSE at a shallower tree depth of 1, highlighting differences in model complexity.

Tree Depth	Crime Inclusive MSE	Crime Exclusive MSE
1	37.94	
2	37.94	
3	38.22	161.56
4	36.13	162.00
5	37.50	162.73

Table 7: Results of Tuning for Tree Depth

Crime-inclusive model achieved its lowest MSE of 36.99 with 1000 trees, demonstrating the importance of adequate tree numbers for model performance. The crime-exclusive model showed less sensitivity to the number of trees, with relatively stable MSE values.

Number of Trees	Crime Inclusive MSE	Crime Exclusive MSE
100	37.03	153.29
500	37.42	152.80
1000	36.99	152.31
10000	37.06	152.31

Table 8: Results of Tuning for number of trees

The crime-inclusive model achieved its lowest MSE of 26.50 with an MTRY of 10, indicating the importance of feature selection in optimizing model performance. In contrast, the crime-exclusive model showed its lowest MSE at an MTRY of 9.

MTRY	Crime Inclusive MSE	Crime Exclusive MSE
10	26.50	-
9	27.52	148.46
8	28.85	153.54
7	30.46	155.61

Table 9: Results of Tuning MTRY Parameter

13 Code Appendix

Data Cleaning:

```
rm(list = ls())
install.packages("magrittr")
install.packages("dplyr")
install.packages(’ggplot2’)
install.packages(’lattice’)
install.packages("class")
install.packages("car")
library(car)
library(class)
library(caret)
library(randomForest)
library(readr)
library(dplyr)
library(ggplot2)
library(gridExtra)
library(dplyr)
library(magrittr)
library(readr)


REC_Df <- read_csv("Desktop/Real Estate Crime Rate Philly.csv")

dim(REC_Df)

REC_Df <- REC_Df[complete.cases(REC_Df), ] #has to be done because there
#is 190 rows with no data, in any column other than key.

#remove these as dont know what they mean
REC_Df <- REC_Df[, -which(names(REC_Df) %in% c("Ward", "Seller", "Buyer", "Zillow
    Estimate","Rent Estimate", "finished", "Average comps", "Book/Writ"))]
REC_Df <- REC_Df[, -which(names(REC_Df) %in% c("Sale Date", "OPA", "PropType"))]


head(REC_Df)
########################################################### checking for null
REC_Df <- REC_Df[complete.cases(REC_Df), ]
missing_values <- is.na(REC_Df)
missing_counts <- colSums(missing_values)
print(missing_counts)
#No Na’s found

######################################################## checking for duplicates
(sum(duplicated(REC_Df)))
#NO DUPLICATES

########################################################checking for 0’s
zero_counts_per_column <- apply(REC_Df == 0, 2, sum)

print("Number of 0s in each column:")
print(zero_counts_per_column)

REC_Df <- REC_Df[, -which(names(REC_Df) %in% c("PGW", "Other","Record Deed"))]

dim(REC_Df)

# Example: Remove rows with zeros in the ’Water’ column
REC_Df <- REC_Df[REC_Df$Water != 0, ]
```

```r
REC_Df <- REC_Df[REC_Df$yearBuilt != 0, ]
REC_Df <- REC_Df[REC_Df$Advertising != 0, ]
REC_Df <- REC_Df[REC_Df$‘finished \n(SqFt)‘ != 0, ]

dim(REC_Df)

#####################################################checking for hyphens
#counting hyphens in each column in data set
count_hyphens <- function(column) {
  sum(grepl("-", column))
}
hyphen_counts <- sapply(REC_Df, count_hyphens)
print("Number of hyphens in each column:")
print(hyphen_counts)

#removing rows with - in them , as when found in bed room they are found in
    bathroom
rows_with_hyphens1 <- grep("-", REC_Df$bathrooms)
num_rows_with_hyphens1 <- length(rows_with_hyphens1)
print(paste("Number of rows with hyphens in 'bathroom' column:", num_rows_with_
    hyphens1))
REC_Df <- REC_Df[-rows_with_hyphens1, ]
dim(REC_Df)

# #######################################Remove variables by index using subset
    notation
#remove OPA as have tax column
# Remove columns with spaces in their names using backticks
REC_Df <- REC_Df[, -which(names(REC_Df) %in% c("Zillow Address","Opening Bid", "
    Attorney", "Address" ))]

dim(REC_Df)

########################################################cleaning the response
    varaible


REC_Df <- REC_Df %>%
  rename(house_price= ‘Sale Price/bid price‘)

REC_Df$house_price <- gsub("\\$", "", REC_Df$house_price)
REC_Df$house_price <- gsub(",", "", REC_Df$house_price)


# Convert the "house_price" column from character to numeric
REC_Df$house_price <- as.numeric(REC_Df$house_price)

####################################################### Printing column names
print(names(REC_Df))
head(REC_Df)

REC_Df <- REC_Df %>%
  rename(postcode = ‘Postal Code‘, sheriff_cost = ‘Sheriff Cost‘, walking_transit_
    score= ‘Avg Walk&Transit score‘,
         violent_crime_rate = ‘Violent Crime Rate‘, school_score= ‘School Score‘,
    squarefeet= ‘finished \n(SqFt)‘ )

###############################################Changing Data types
REC_Df$bedrooms <- as.numeric(REC_Df$bedrooms)
REC_Df$bathrooms <- as.numeric(REC_Df$bathrooms)

head(REC_Df)
data_types<- sapply(REC_Df, class)
data_types


###############scaling response variable

# Assuming REC_Df is your dataframe containing the house_price variable
# Scale house_price by multiplying by 1000
REC_Df$house_price <- REC_Df$house_price / 1000
```

```
# Check the first few rows of the updated dataframe
head(REC_Df)
```

Data Analysis:

```
#######corelaation of response variable with all numeric variables in the data set
    .
correlation_with_house_price <- cor(REC_Df[, sapply(REC_Df, is.numeric)],
                                    REC_Df$house_price)

# Print correlation coefficients
print(correlation_with_house_price)

############pairwise
pairs(~ house_price + sheriff_cost + walking_transit_score + violent_crime_rate +
    school_score + taxAssessment,
      data = REC_Df,
      col = "steelblue")
#########histogrM
# Extract house prices from REC_Df
house_prices <- REC_Df$house_price

# Create a histogram plot without title and axis labels
hist(house_prices,
     xlab = "House Price (in thousands)",
     ylab = "Frequency",
     col = "steelblue",
     border = "black",
     breaks = 20,
     main = "",
     axes = TRUE,
     ylim= c(0,200))

# Calculate the mean of house prices
# Extract house prices from REC_Df
house_prices <- REC_Df$house_price

mean_price <- mean(house_prices)
# Add a vertical line at the mean value
abline(v = mean_price, col = "red", lwd = 2, lty = 2)


##########var importance
# Fit random forest model
set.seed(123)  # Set seed for reproducibility
rf_model <- randomForest(house_price ~ ., data = REC_Df)

# Extract variable importance scores
importance_scores <- importance(rf_model)

# Create data frame for plotting
importance_df <- data.frame(
  Variable = rownames(importance_scores),
  Importance = importance_scores[, "MeanDecreaseGini"]
)

# Sort data frame by importance values (ascending order)
importance_df <- importance_df[order(importance_df$Importance), ]

# Define colors for top and bottom variables
importance_df$Color <- ifelse(
  importance_df$Variable %in% head(importance_df$Variable, 2),
  "red", "steelblue"
)

# Create a horizontal bar plot for variable importance
barplot(importance_df$Importance, horiz = TRUE,
        names.arg = importance_df$Variable,
        col = importance_df$Color,
        xlab = "Importance",
        xlim = c(0, max(importance_df$Importance) * 1.1),
        las = 1,  # Rotate names of variables
        border = NA,
        ylab= 'varaibel')  # Remove border around bars
```

Model:

```

set.seed(100)

# Define the list of variables you want to include in your model
variables_of_interestrg1 <- c(
  "house_price",
  "sheriff_cost",
  "taxAssessment",
  "school_score",
  "walking_transit_score",
  "squarefeet",
  "violent_crime_rate",
  "postcode",
  "yearBuilt",
  "bathrooms",
  "Advertising"
)

# Sample splitting proportion
train_proportion <- 0.8
train_size <- round(nrow(REC_Df) * train_proportion)

# Create training and test datasets
train_indices <- sample(seq_len(nrow(REC_Df)), size = train_size, replace = FALSE)
train_datarg1 <- REC_Df[train_indices, variables_of_interestrg1]
test_datarg1 <- REC_Df[-train_indices, variables_of_interestrg1]

# Specify parameters for the random forest model
ntree <- 750
maxdepth <- 4
mtry <- 9

# Initialize an empty vector to store MSE scores
mse_vector1 <- numeric()

# Define the number of iterations or runs
num_iterations <- 10  # You can adjust this based on your needs
num_folds <- 10       # Number of folds for cross-validation

# Create indices for k-fold cross-validation
folds <- createFolds(y = train_datarg1$house_price, k = num_folds)

set.seed(100)

for (fold in 1:num_folds) {
  # Extract training and validation sets for the current fold
  train_indices <- unlist(folds[-fold])
  validation_indices <- unlist(folds[fold])

  train_data1 <- train_datarg1[train_indices, ]
  validation_data1 <- train_datarg1[validation_indices, ]

  # Train the random forest model
  rg1 <- randomForest(
    formula = house_price ~ .,
    data = train_datarg1,
    ntree = 1000,
    mtry = 10,
    maxdepth = 4
  )

  # Make predictions on the validation set
  predicted1 <- predict(rg1, newdata = validation_data1)

  # Extract the actual house prices from the validation set
  actual1 <- validation_data1$house_price
  # Calculate Mean Squared Error (MSE) for the fold and store it
  mse_fold <- mean((actual1 - predicted1)^2)
  mse_vector1 <- c(mse_vector1, mse_fold)
}

# Calculate the average MSE across all folds
average_mse <- mean(mse_vector1)
mse_vector1






#####################################################vector of MSE2 - NAviolence
set.seed(100)
library(randomForest)
library(caret)

variables_of_interestrg2 <- c(
  "house_price",
  "sheriff_cost",
  "taxAssessment",
  "school_score",
  "walking_transit_score",
  "squarefeet",
  "postcode",
  "yearBuilt",
  "bathrooms",
  "Advertising"
)

train_indices <- sample(seq_len(nrow(REC_Df)), size = train_size, replace = FALSE)
train_datarg2 <- REC_Df[train_indices, variables_of_interestrg2]
test_datarg2 <- REC_Df[-train_indices, variables_of_interestrg2]

# Initialize an empty vector to store MSE scores
mse_vector2 <- numeric()

# Define the number of iterations or runs
num_iterations <- 10  # You can adjust this based on your needs
num_folds <- 5     # Number of folds for cross-validation

# Create indices for k-fold cross-validation
folds <- createFolds(y = train_datarg2$house_price, k = num_folds)

for (fold in 1:num_folds) {
  # Extract training and validation sets for the current fold
  train_indices <- unlist(folds[-fold])
  validation_indices <- unlist(folds[fold])

  train_data <- train_datarg2[train_indices, ]
  validation_data <- train_datarg2[validation_indices, ]

  # Train the random forest model
  rg2 <- randomForest(
    formula = house_price ~ .,
    data = train_data,
    ntree = 10000,
    mtry = 9,
    maxdepth = 3
  )

  # Make predictions on the validation set
  predicted <- predict(rg2, newdata = validation_data)

  # Extract the actual house prices from the validation set
  actual <- validation_data$house_price

  # Calculate Mean Squared Error (MSE) for the fold and store it
```

```r
  mse_fold <- mean((actual - predicted)^2)
  mse_vector2 <- c(mse_vector2, mse_fold)
}

# Calculate the average MSE across all folds
average_mse <- mean(mse_vector)

# Print or view the average MSE
print(average_mse)

mse_vector2
mse_vector1


################################################################making my vectors
# Set seed for reproducibility
set.seed(100)

# Define variables of interest for both models
variables_of_interest <- c(
  "house_price",
  "sheriff_cost",
  "taxAssessment",
  "violent_crime_rate",
  "school_score",
  "walking_transit_score",
  "squarefeet",
  "postcode",
  "yearBuilt",
  "bathrooms",
  "Advertising"
)

# Sample splitting proportion
train_proportion <- 0.8
train_size <- round(nrow(REC_Df) * train_proportion)

# Create indices for consistent sampling
train_indices <- sample(seq_len(nrow(REC_Df)), size = train_size, replace = FALSE)

# Initialize empty vectors to store MSE scores
mse_vector11 <- numeric()
mse_vector2 <- numeric()

# Define number of iterations and consistent number of folds
num_iterations <- 10
num_folds <- 10

# Create indices for k-fold cross-validation
folds <- createFolds(y = REC_Df$house_price, k = num_folds)

# Loop over folds for model training and evaluation
for (fold in 1:num_folds) {
  # Extract training and validation indices
  train_indices <- unlist(folds[-fold])
  validation_indices <- unlist(folds[fold])

  # Subset data for both models based on indices
  train_data <- REC_Df[train_indices, variables_of_interest]
  validation_data <- REC_Df[validation_indices, variables_of_interest]

  # Train and evaluate model 1 (rg1)
  rg1 <- randomForest(
    formula = house_price ~ .,
    data = train_data,
    ntree = 1000,
    mtry = 10,
    maxdepth = 4
  )
  predicted11 <- predict(rg1, newdata = validation_data)
  actual11 <- validation_data$house_price
  mse_vector11 <- c(mse_vector11, mean((actual11 - predicted11)^2))

  # Train and evaluate model 2 (rg2)
  rg2 <- randomForest(
    formula = house_price ~ . -violent_crime_rate,
    data = train_data,
    ntree = 1000,
    mtry = 9,
    maxdepth = 3
  )
  predicted2 <- predict(rg2, newdata = validation_data)
  actual2 <- validation_data$house_price
  mse_vector2 <- c(mse_vector2, mean((actual2 - predicted2)^2))
}

# Print or view MSE vectors
mse_vector1
mse_vector2

# Calculate and print average MSE for each model
average_mse1 <- mean(mse_vector11)
average_mse2 <- mean(mse_vector2)
print(paste("Average MSE for Model 1 (rg1):", average_mse1))
print(paste("Average MSE for Model 2 (rg2):", average_mse2))
```

Results:

```
mse_vector1
mse_vector2

average_mse1 <- mean(mse_vector1)
average_mse1
average_mse2
average_mse2 <- mean(mse_vector2)
print(paste("Average MSE for Model 1 (rg1):", average_mse1))
print(paste("Average MSE for Model 2 (rg2):", average_mse2))

result <- t.test(mse_vector2, mse_vector1, paired = TRUE)
print(result)

# Interpret the result
if (result$p.value < 0.05) {
  print("There is a significant difference in mean MSE scores.")
} else {
  print("There is no significant difference in mean MSE scores.")
}

library(ggplot2)

# Define the sequence of data points (e.g., scenarios or datasets)
data_points <- 1:length(MSE_with)

# Create a data frame with MSE values and corresponding data points for each model
data <- data.frame(
  Data_Point = data_points,
  MSE_Crime_Inclusive = MSE_with,
  MSE_Crime_Exclusive = MSE_without
)

# Reshape the data frame into long format for ggplot
data_long <- tidyr::gather(data, Model_Type, MSE, -Data_Point)

# Plotting the line graph
ggplot(data_long, aes(x = Data_Point, y = MSE, color = Model_Type)) +
  geom_line(size = 1) +
  labs(
    x = "Data Point",
    y = "Mean Squared Error (MSE)",
    color = "Model Type"
  ) +
  geom_hline(yintercept = 15.90, linetype = "dashed", color = "steelblue") +
  geom_hline(yintercept = 96.00, linetype = "dashed", color = "#FF5733") +
```

```r
  theme_minimal() +
  coord_cartesian(xlim = c(0, 12)) +  # Set x-axis limits using coord_cartesian
  theme(
    axis.text.x = element_text(color = "black", size = 12),
    axis.text.y = element_text(color = "black", size = 12),
    axis.title = element_text(color = "black", size = 14),
    legend.position = "top"
  )
```