To What Extent Did 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 because VCRs 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 existing literature by empirically investigating the impact of violent crime rates on house prices, aiming to provide new insights into this relationship.Through the analysis of Pennsylvania house price data and the 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. This includes proposing empirically supported strategies to address violent crime and inform evidence-based policymaking 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 VCRs since 2014 (Berk, 2022) highlights 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 neighbourhood dynamics such as disinvestment, demolition, and deindustrialization, can accelerate neighbourhood decline by weakening social controls and impacting urban development outcomes (Skogan, 1986). Areas with higher VCRs 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 VCRs 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 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 modelling effectiveness.

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

Table 1: Percentage of Zero Values in Selected Variables

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

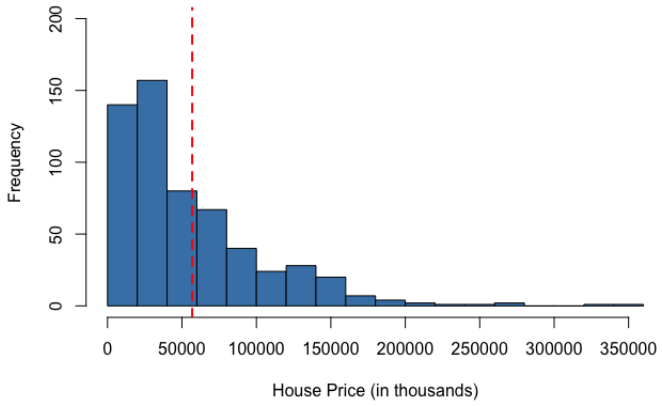
Data Classifications	Features
Categorical Data	Postcode, Year the House was Build
Discrete Data	Number of Bedrooms, Number of Bathrooms
Continuous Data	Sheriff Cost, Advertising, Water, Walking Score, Violent crime rate, School Score, Tax Assessment

Table 2: Variable Data Types

3.3 Data Exploration

Figure 1 illustrates house prices ranging from $5,200 to $3,500,000. While the median and mode are both $40,000, which is lower than the mean ($57,772). When higher-priced houses extend beyond the red dashed line, representing the mean, it influences the distribution, resulting in a mean that surpasses the median and mode. This highlights the asymmetry of the house price distribution, with a notable concentration of prices clustering towards the lower end, contributing to the observed leftward skewness.

Figure 1: **Distribution of House Prices**



Outliers, which are values that exceed 1.5 times the interquartile range, are detected in 2.9% of the observations in the response variable based on outlier tests. Figure 1, in conjunction with the analysis of central tendencies for house prices and further assessment of these observations, suggests that they are valid and do not raise any concerns. Therefore, will be retained in the dataset. (Refer to Appendix 2 for additional assessment details.)

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

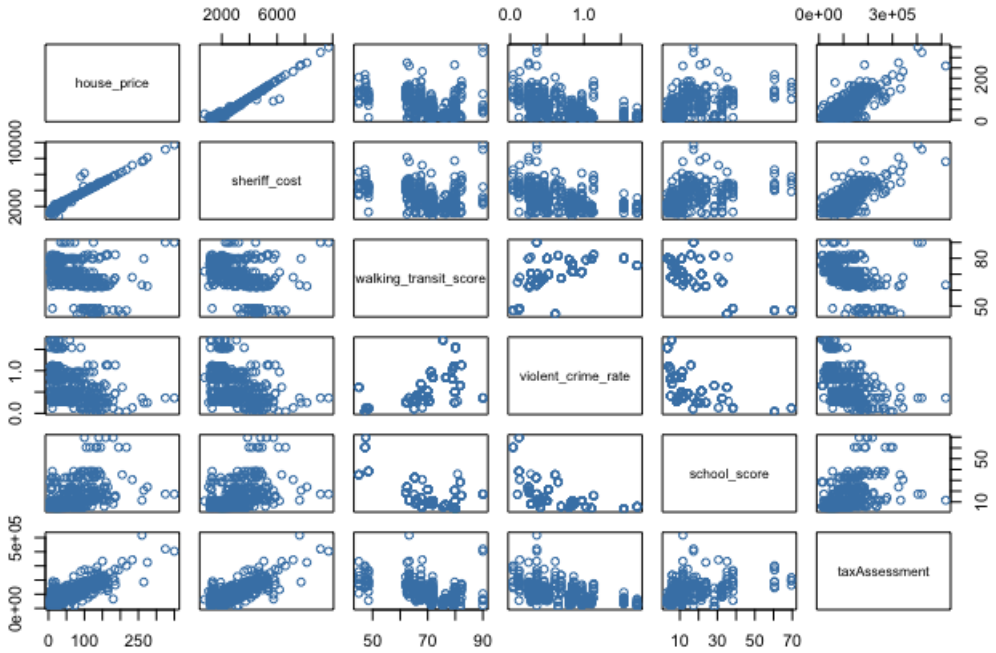
Table 3 presents correlation coefficients between the response variable house price and all explanatory variables, highlighting notably strong correlations with sheriff cost and tax assessment. Concerned about these high correlations, an investigation into potential multicollinearity involved examining the top 5 correlated variables with each other and with the response variable.

To assess multicollinearity, variance inflation factor (VIF) scores were computed, shown in Table 4. Importantly, all VIF scores are below 5, indicating no significant multicollinearity 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. Additionally, there was no multicollinearity observed between the variables themselves.

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: VIF of Predictor Variables in Relation to House Price

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



In Figure 2, key variables reveal significant patterns. Sheriff cost correlates positively with House Price, suggesting higher prices in areas with elevated sheriff costs, while longer commute distances are linked to lower prices. Higher house prices coincide with lower VCRs, indicating safer neighbourhoods in affluent areas. School scores show consistency across house prices, suggesting limited impact on housing dynamics. Tax assessment weakly correlates 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 a random forest regression model, a method well-suited for analysing house prices. This approach is effective for this dataset due to many buyers following a checklist of amenities when deciding on a house, such as crime rates or commute time. This model is adept at accommodating diverse variable types, aligning well with the varied classifications indicated in the dataset. Furthermore, this model is robust against outliers and effectively handles datasets with approximately 500 observations. This method splits the solution space with cut-offs of the explanatory variables, providing insights into the relationships between different factors influencing house prices, crucial for understanding the link between house prices and VCRs. This approach is well-suited to this dataset with diverse correlations, as it can capture complex relationships and identify important features contributing to house prices, which multiple linear regression lacks the ability to do effectively. Additionally, the combination of decision trees in random forest improves generalisation performance, enhancing the reliability of this model type and reducing the 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 illustrates that employing a random forest model with 10 predictors achieves an optimal balance between complexity and performance, thereby improving the model's effectiveness and reliability. In Figure 2, water and bedrooms, which exhibit correlations of less than 0.1 with house price, as shown in the data section, are recognized as the two least significant variables and will be dropped from the model.

Figure 3: Optimal Predictor Count and Test MSE Relationship

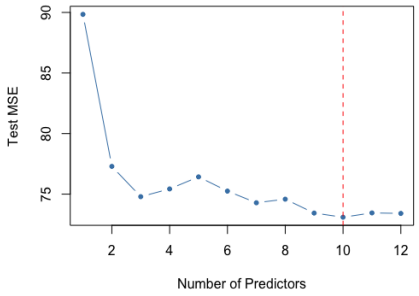
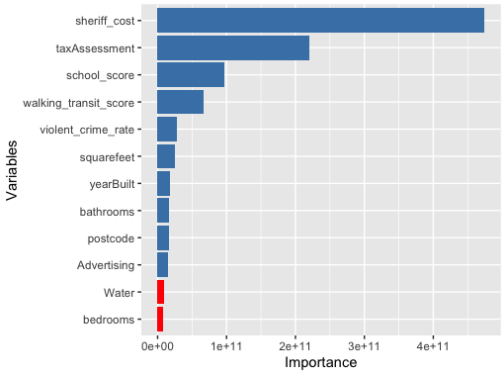


Figure 4: Variable Importance in Random Forest



Two random forest models were constructed to explore the effect of VCRs on house prices—one including this predictor and another excluding it. This method directly evaluates the effect of VCRs on house prices and facilitates a comparative analysis to quantify their specific contribution.

- Random Forest 1 (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 2 (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:

$$\begin{aligned}
\hat{Y}_1 &= \text{predicted house prices for model with violent crime rate} \\
\hat{Y}_2 &= \text{predicted house prices for model without violent crime rate} \\
f &= \text{random forest prediction function} \\
X_1 &= \text{violent_crime_rate} \\
X_2 &= \text{walk_transit_score} \\
X_3 &= \text{school_score} \\
X_4 &= \text{tax_assessment} \\
X_5 &= \text{sheriff_cost} \\
X_6 &= \text{advertising} \\
X_7 &= \text{year_built} \\
X_8 &= \text{squarefeet} \\
X_9 &= \text{bathrooms} \\
X_{10} &= \text{postcode}
\end{aligned}$$

The optimal number of K-folds (cross-validation) for the two random forest models are 5 K-folds for the VCR-inclusive model and 10 K-folds for the VCR-exclusive model. This variation emphasizes the significance of the VCR predictor in improving model accuracy and highlighting the need for tailored cross-validation strategies to optimize predictive performance based on model configuration.

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

The optimal number of trees in the crime-inclusive model is 10,000, compared to 1,000 in the crime-exclusive model. This contrast highlights the importance of discerning how the inclusion of VCRs distinctly shapes model performance under varying conditions, further emphasizing the variable's significance in predictive accuracy.

The optimal number of randomly selected predictors considered at each split (MTRY) in both models is where the number of predictive variables is maximised. The crime-inclusive model is optimized at MRTY=10 and the crime-exclusive model is optimized at MRTY=9. Therefore, optimizing MTRY at both models maximum MTRY setting achieves better generalisations to unseen data, particularly when considering the variable inclusion model.

These random forest regression models provide robust frameworks to examine how the inclusion of the VCRs impact 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. (See Appendix 3 for detailed statistical tables corresponding to this analysis.)

5 Results

5.1 Model Performance Metrics

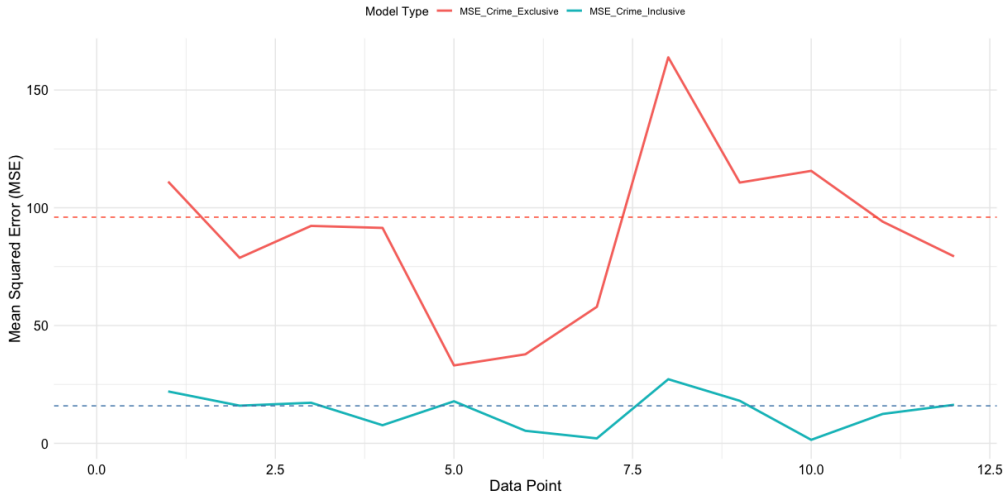
Model Metric	Random Forest Crime-Inclusive	Random Forest Crime-Exclusive
MSE	15.90	96.00
RMSE	3.99	9.80
MAE	1.48	3.06
R-Squared	0.97	0.82

Table 5: Model Performance Metric Results

Table 5 highlights the outstanding predictive performance of the crime-inclusive model (MSE = 15.90), which is six times more accurate than the crime-exclusive model (MSE = 96.00), highlighting the importance of incorporating crime rates as a predictor in understanding neighbourhood dynamics and housing market behaviour.

The lower MSE of the crime-inclusive model reflects its enhanced accuracy in capturing market complexities and buyer preferences in house price 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 5.

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



The RMSE of approximately $3,990 for the crime-inclusive model indicates that, on average, predictions deviate by this amount from the actual house prices. Comparatively, this score suggests better accuracy than the model excluding crime, indicating good precision for both models. A low RMSE relative to house prices signifies minimal prediction errors, suggesting that most underlying factors are captured by the models. However, the complexity of accurately predicting house prices is highlighted by the recognition that despite achieving good precision, challenges remain due to the multitude of factors influencing housing markets.

The crime-inclusive model exhibits a lower MAE of 1.48, indicating that its predictions in absolute terms are closer to actual house prices compared to the crime-exclusive model, which has an MAE of 3.06. This difference emphasizes the accuracy and magnitude of prediction error of the crime-inclusive model, highlighting its better predictive performance in capturing house price variability.

With an impressive R-squared value of 97%, the crime-inclusive model demonstrates 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 test 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 emphasizes the crucial role of VCRs in predicting house prices and highlights 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 was conducted using MSE values obtained from both the crime-inclusive and crime-exclusive models. This approach allows for assessing how the inclusion of violent crime rates influences the accuracy of our house price prediction model.

- 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 conducted to compare the distribution of MSE values between the crime-inclusive model and the crime-exclusive model. The resulting p-value of 0.04635 indicates a statistically significant difference in mean MSE values between the two models. With a significance level set at 0.05, the observed p-value falls below this threshold, leading to the rejection of the null hypothesis.

- Given:

$$\alpha = 0.05$$

$$p = 0.04635$$

- Statistical Decision:

$$0.04635 < 0.05$$

This rejection suggests that including VCRs 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 violent-crime data into predictive modelling to better understand neighbourhood dynamics and house pricing behaviour.

6 Discussion

These results highlight the substantial impact of VCRs 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 by 1.58 compared to the crime-exclusive model.

The hypothesis testing that used 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. This highlights the influence of violent crime rates on house price predictions and emphasizes the critical role of violent 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 price stability.

This approach facilitates the development of effective strategies to address crime, attract investments, and improve overall neighbourhood well-being. Moreover, incorporating crime data in policymaking ensures that interventions are targeted and evidence-based, thereby nurturing community well-being and building resilient communities.

7 Limitations

The analysis is constrained by the relatively small dataset, potentially limiting the generalizability of findings. Utilizing a larger dataset could offer more comprehensive insights and yield robust conclusions. Additionally, while outliers were manageable, their presence may have distorted the analyses. Future studies with larger datasets could ensure a more accurate representation of the relationship between violent crime rates and house prices by identifying and removing outliers.

Another limitation of this analysis is the potential occurrence of omitted variable bias, where relevant factors influencing the relationship between violent crime rates and house prices are not accounted for. The exclusion of relevant socioeconomic or demographic variables, such as unemployment rates, could affect the accuracy and reliability of predictive model outputs.

Accurately assessing the importance of the predictor variable sheriff cost in the random forest model is challenging due to its significant correlation with house prices. This may lead to an incomplete understanding of its significance. Addressing this issue in future studies is crucial for a clearer interpretation of the role of sheriff costs in predicting house prices. Also failing to account for this could also affect the perceived importance of other variables in the model

Furthermore, the analysis focuses on model refinement and complexity highlighting the need for additional variables. These variables could enhance the predictive capabilities of the models, especially in understanding the complex relationship between VCRs and house prices. Moreover, the model's use of the maximum number of MTRY may not be optimal, suggesting that more informative predictors could improve model performance.

8 Policy Implications

The implementation of Crime Prevention Through Environmental Design (CPTED) grants would be a pivotal strategy in mitigating VCRs and catalysing positive house prices in Pennsylvania. This policy aims to provide financial assistance to multiple groups, such as property owners and city councils, to install CPTED features such as landscape improvements, signalling safety, and deterring crime in these revitalized areas (Ha, Oh, & Park, 2015).

These grants aim to enhance community safety by installing CPTED features like security cameras and street lighting, effectively reducing crime rates and increasing perceived safety (Kim & Park, 2017). Consequently, this reduction would increase demand for houses in these neighbourhoods, leading to higher property values. By altering the physical environment to discourage criminal activity, this policy design not only reduces crime rates but also enhances the desirability of these neighbourhoods, attracting home buyers willing to pay higher prices for homes in safer communities (Ha, Oh, & Park, 2015).

The proposed CPTED grant program aligns with the overarching goal of tackling the complex challenges of crime prevention and house prices in Pennsylvania. By nurturing community well-being, creating resilient communities, and addressing immediate safety concerns, laying the foundation for sustained revitalization and enduring growth (Johnson, Gibson, & Stevens, 2014).

9 Conclusion

In summary, this essay provides actionable insights into the negative impact of violent crime rates on house prices in Pennsylvania in 2016, emphasizing the significance of incorporating crime data in housing market analysis for evidence-based policymaking and interventions aimed at nurturing safer and more prosperous communities.

10 Bibliography

1. Ahmad, A., & Abubakar, A. (2021). *VIOLENT CRIME PERCEPTION AND REAL ESTATE PRICES: RELATED EMPIRICAL STUDIES IN PERSPECTIVE. Creative Business Research Journal*, 1(2), 106–107. https://doi.org/2756-4932

2. Berk, R. (2022). *Is Violent Crime Increasing?* Department of Criminology. Retrieved March 23, 2024, from 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. Ha, T., Oh, G.-S., & Park, H.-H. (2015). *Comparative analysis of Defensible Space in CPTED housing and non-CPTED housing. International Journal of Law, Crime and Justice*, 43(4), 496–511. https://doi.org/10.1016/j.ijlcj.2014.11.005

5. 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

6. Johnson, D., Gibson, V., & Stevens, E. (2014, May 15). *Developing & maintaining sustainable communities: Managing the output focus of Crime Prevention through Environmental Design (CPTED).* Retrieved April 26, 2024, from https://nrl.northumbria.ac.uk/id/eprint/16547

7. Kim, D., & Park, S. (2017). *Improving community street lighting using CPTED: A case study of three communities in Korea. Sustainable Cities and Society*, 28, 233–241. https://doi.org/10.1016/j.scs.2016.09.016

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

9. Pope, D., & Pope, J. (2011). *Crime and property values: Evidence from the 1990s crime drop. Elsevier*, 2(2), 185–197. https://doi.org/177%E2%80%93188

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

11. The Pennsylvania Statistical Analysis Center. (2019). *Pennsylvania a report of crime trends and statistics 2012-2016*, 1–6. Retrieved from https://www.pccd.pa.gov/Justice-Research/Documents/PA%20Crime%20Trends%202012-2016.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.

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

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

```
# 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.

```
# 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.

```
# 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

Further assessment was conducted on the response variables identified as outliers by iterating through the observations and verifying whether other variables, such as the number of bedrooms and bathrooms, aligned with those typically found in more expensive houses. Moreover, examination of additional factors, such as schooling scores for all detected outliers, revealed that at least two or more other variables exhibited characteristics consistent with the price of these houses.

13 Appendix 3

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

14 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, ]
```

```
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
```

```
  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") +
```

```
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"
)
```