Predicting Property Values in Philadelphia

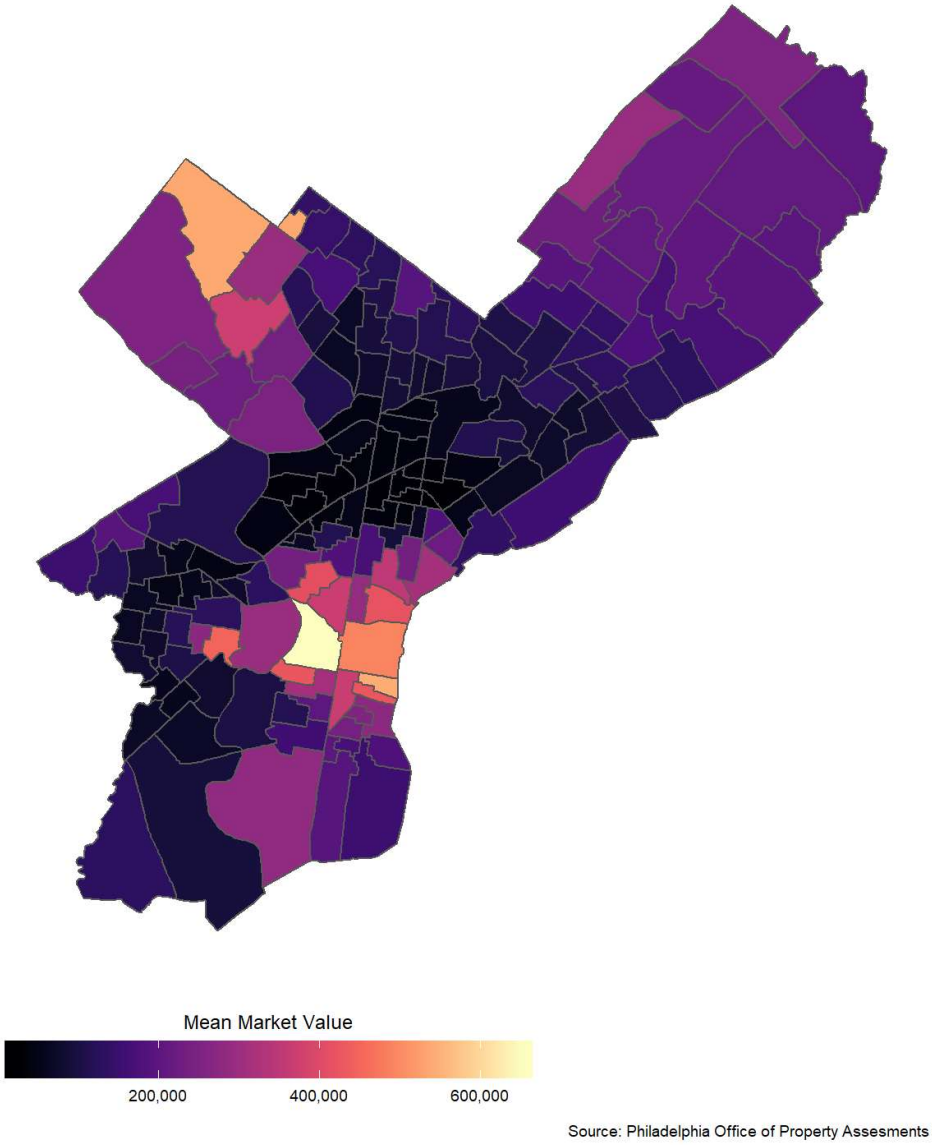
Amar Adusumilli

2019-12-20

Introduction

Following decades of decline wrought by de-industrialization, population loss, and other factors, the city of Philadelphia has shown tangible signs of revival, evidence of which is reflected through quantitative data. The population of the city has increased in every year since 2006, and strong sales in the housing market indicate that this trend is likely to continue. The city's most recently reported unemployment rate was 5.2% - a level unseen since the winter of 2000 (Bureau of Labor Statistics, 2019). The jail population has decreased by 39.5% since July 2015, the product of significant reform in the criminal justice system (MacArthur Foundation, 2019). The public high school graduation rate sits approximately 12% higher than it did in 2009 (Pew, 2019). Many other figures can be cited as indicators of positive change. However, with the third highest income inequality among American cities (Bloomberg, 2018), progress and growth has not been disseminated equally. Population growth has been largely situated in the center of city (Center City) and adjoining neighborhoods, with other outlying areas showing little change or continued declines in some cases. Median household income is only higher than the national average in approximately 25% of the city. In a similarly sized area, median household income is less than half of that mark (Pew, 2019). This inequality is best visualized through geography.

Figure 1: Property Values in Philadelphia
Average Market Value by Elementary School Catchment Area, 2019



Source: Philadelphia Office of Property Assesments

The geographic distribution of single family home values shows a great dichotomy between the areas proximate to Center City and those further north and west. Distributions of educational attainment, crime, median income, and life expectancy show comparable geographic variation. Similar trends can be seen in many other cities across the United States, with varying levels of severity. While the reasons for this variation are contextually dependent between cities, and indeed, between neighborhoods within cities, certain shared factors inform the inequality visible on the map above. As mentioned in the opening sentence of this paper, de-industrialization and population loss were major reasons for Philadelphia's decline, but they too were not distributed evenly across the city. The dark shaded areas on the map correspond to the places most severely afflicted by these events, neighborhoods concentrated within the North and West sections of Philadelphia. Note that this does not apply to the regions to the far Northwest and Northeast - these neighborhoods are more suburban in character, with different patterns of development compared to the rest of the city. A full analysis of the multi-faceted causes and ramifications of post-industrial decline is beyond the scope of this paper, but it is worth noting for the clear impact said decline continues to have on market values throughout the city. I surmise that there must be some basket of explanatory variables which account for the low market values present in North and West Philadelphia, and conversely, for the high values present in Center City. In this paper, I attempt

to predict single family property values in Philadelphia utilizing regression analysis. Through this process, I aim to assess the importance of the explanatory variables and present considerations to potentially direct and improve further analysis.

Hypothesis

In linear regression, for any variable j the null hypothesis is that the coefficient $\beta_j = 0$. The alternative hypothesis is that the coefficient $\beta_j \neq 0$. In more succinct notation, we can say: $H_0 : \beta_j = 0;\ H_a : \beta_j \neq 0$. To assess significance, we use t-statistics, calculated under the general form:

$$t_j = \frac{\beta_j - 0}{SE_{\beta_j}}$$

If the coefficient for variable j is 0, we conclude that variable j has no effect on the dependent variable. This is simple to see mathematically - if the coefficient of a term is 0, then the value of the term itself is 0, and by extension the term does not contribute to the final prediction of the model. The t-test assesses only one coefficient at a time. To assess the significance of multiple coefficients jointly, the F-test is used. The F-test for linear regression compares a model with no explanatory variables (intercept only) to the specified model. The hypotheses for this test are as follows: $H_0 : \beta_1 = \beta_2 = \ldots = 0;\ H_a : \beta_i \neq 0$. In plain English, the null hypothesis for the F-test is that none of the explanatory variables help explain the dependent variable. The alternative hypothesis is that at least one of the explanatory variables helps to explain the dependent variable.

Data

The primary data used in this analysis was the property assessments data set, published annually by the Philadelphia Office of Property Assessment and publicly available on the Open Data Philly webpage. It contains the officially assessed market value of every property in the city, as well as detail on the size, condition, location, and other attributes of said properties. When filtered on single family homes, the data contains over 450,000 observations and 77 variables. The properties dataset represents the population and contains many idiosyncrasies typical of municipal data. By extension, feature selection and data cleaning were difficult endeavors. The majority of these 77 variables were removed during exploratory data analysis. Variables were either included or dropped based on a somewhat subjective assessment of quality. In general, if a given variable contained a significant proportion of missing values, or if it lacked a comprehensible description in the metadata, it was dropped. This resulted in dropping several variables that would otherwise appear to be relevant, including number of bathrooms and garage spaces. Other variables, such as the name of the property owner, or the order that licensing documents were received, were dropped due to clearly being extraneous to the purpose of predicting market values. Observations were dropped based on a similar criteria - all observations with missing market values were removed, as well as missing values within any of the four independent variables taken from the properties dataset. Additionally, all instances of public housing were removed, as they do not conform to the same market forces as other properties. Similarly, observations corresponding to ‘residential air rights’ and ‘condo parking space’ were removed. Other observations were dropped situationally through the examination of outliers - many were clear errors within data entry or data collection. In all, some 10,000 observations were dropped, leaving 453,367 data points in the final, cleaned dataset.

Additional data used in this analysis pertained to crime and elementary schools. The crime data consisted of daily arrests by police district, spanning back to 2014. This dataset is maintained by the Philadelphia District Attorney’s Office and is updated regularly. Arrests are split out into 18 distinct bins, ranging from homicides to

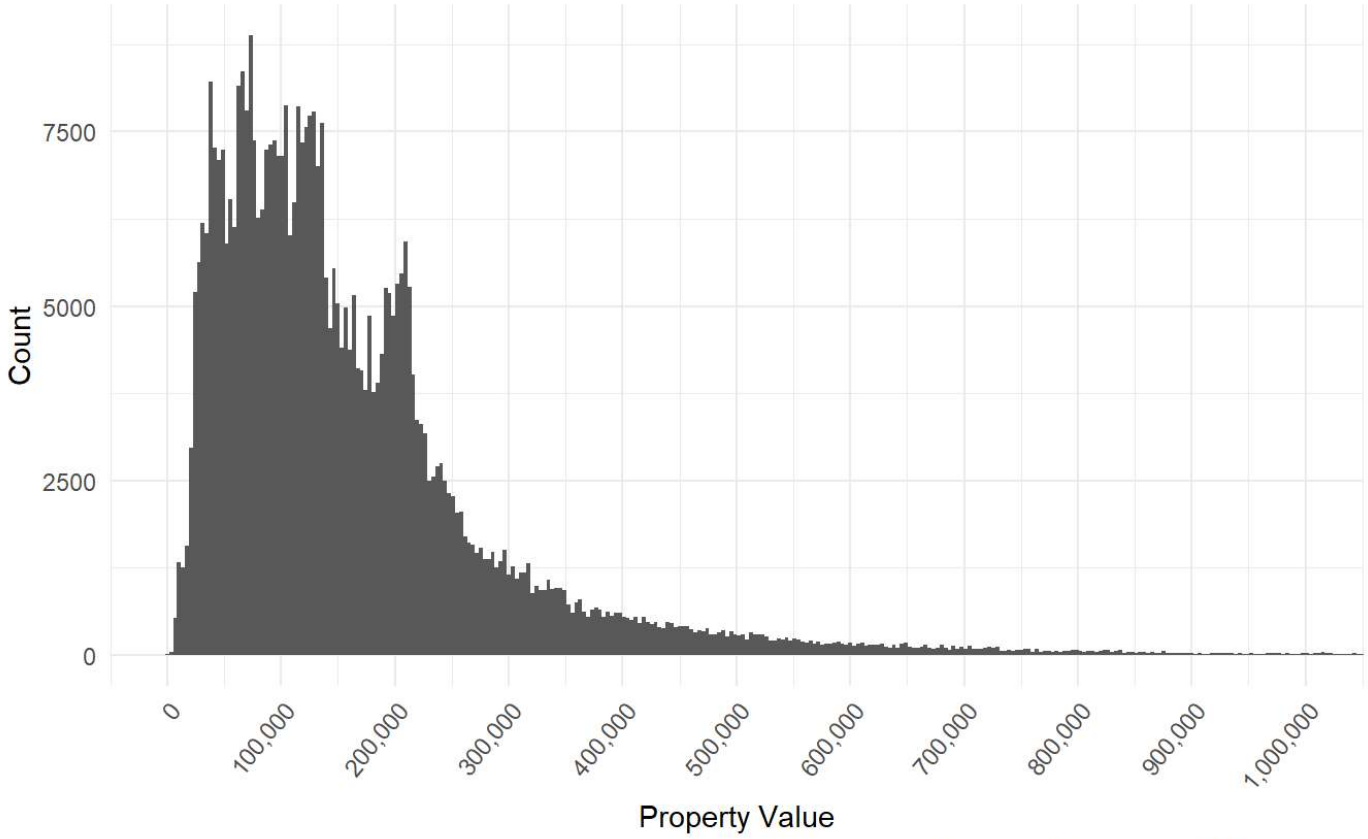
DUI's. The school data, published by the School District of Philadelphia, consisted of numerous progress scores given to the individual elementary schools on various criteria. The 'overall score', an aggregate of these varying ratings, was taken as the measure of school performance. The information contained within these two datasets was added to the properties data through the use of certain Geographic Information Systems (GIS) tools - the specifics are detailed in the 'method' section of this paper.

Method

The core intuition that guided the model building process was that the market value of any given property was a function of the descriptive attributes of the property itself, the characteristics of its surrounding location, and its proximity towards places of interest. Utilizing regression techniques, I attempted to codify this equation. Two models were used: a multiple linear regression model, and a random forest model. The distribution of market values was highly skewed, as seen below:

Figure 2: Distribution of Home Values in Philadelphia

Single Family Homes Only

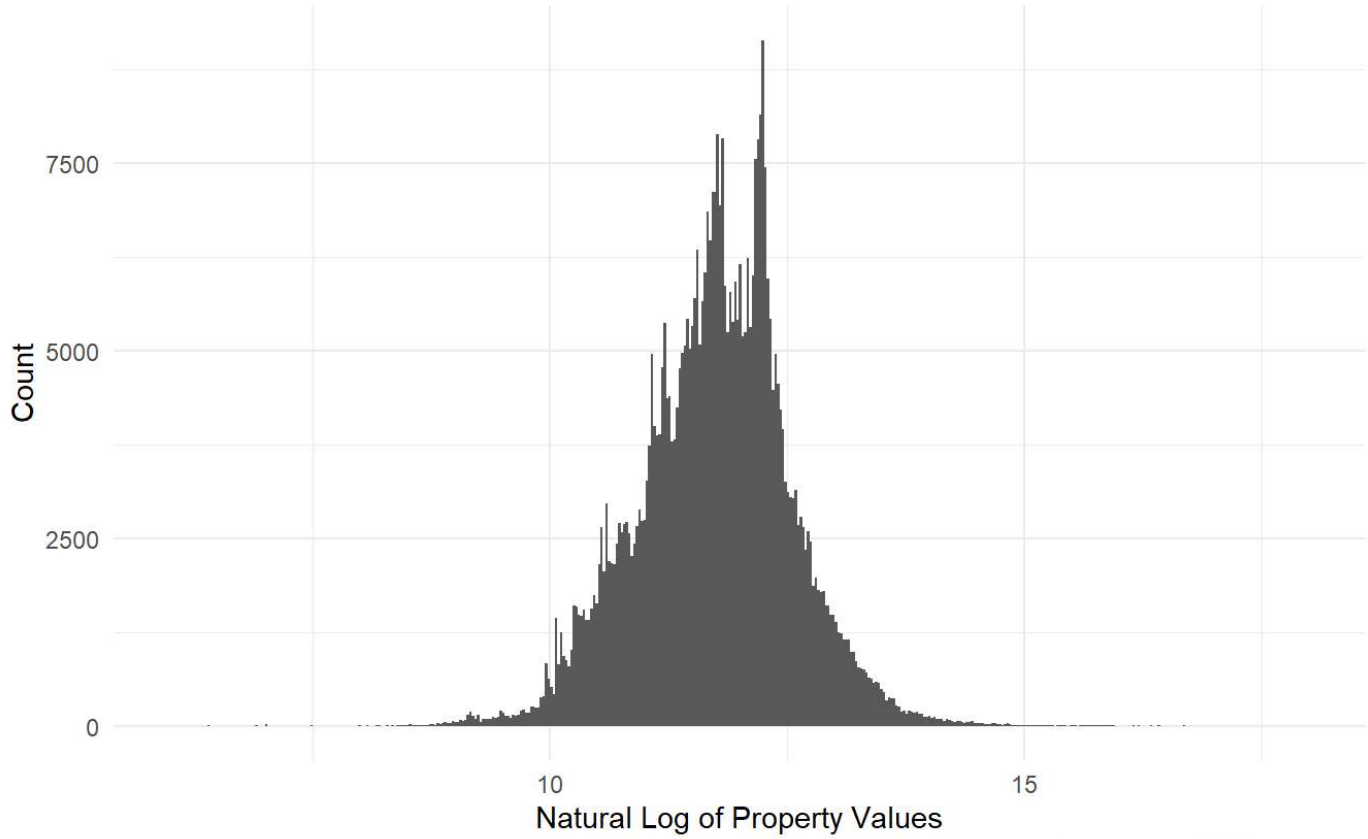


Source: Philadelphia Office of Property Assessment

Due to this skew, a log transformation was used on the distribution of market values. The distribution of the log-transformed market values is far more symmetric, as seen below:

Figure 3: Log Transformed Distribution of Philadelphia Property Values

Single Family Homes Only



Source: Philadelphia Office of Property Assessment

As such, log-transformed market values was the response variable in the regression models. The primary multiple linear regression model contained 9 variables: Natural Log of Total Livable Area (square footage), Interior Condition, Exterior Condition, View Type, Distance to City Hall, Type 1 Crime, Type 2 Crime, Elementary School, and School Performance Score. The natural log of the total livable area was used to smooth an quasi-exponential relationship between market value and total livable area. Interior and exterior condition were rated on scales of 7 to 1, with 7 representing excellent condition and 1 representing severe blight, indicative of visible structural damage or open exposure to the elements. View type was a categorical variable that indicated the type of view that a property has, such as whether it has a skyline view or faces a park. The distance to city hall variable was added to account for proximity. It was calculated using the Haversine formula, which determines the great circle distance between two points on a sphere given their latitudes and longitudes. City Hall was chosen as a proxy for Center City due its central location in Philadelphia's spatial layout. The distance calculation does not account for street grids, which resulted in a systemic underestimate of the true distance. I believe this justifiable because the relative distance between properties was the matter of importance, not the absolute distance to the city center. Type 1 and Type 2 crimes were a weighted daily average of the count of crimes spanning from January 2014 to November 2019, the split made according to the FBI's Uniform Crime Reports criteria. The elementary schools were a categorical variable consisting of every public, catchment admit elementary school within Philadelphia. Private, charter, and special-admit schools were not included. This variable contains 160 levels and as such it forms a way to partition the city into 160 heterogeneous pieces. This, along with the crime variables, was intended to proxy the effect of an area on a property's market value. School score was the 'overall score' mentioned above, an aggregate of various performance ratings that the schools received. The incorporation of the school and crime data was done through the use of the point in polygon algorithm from the R GIS package 'sf'. Every observation in the properties dataset contained latitude and longitude coordinates, and through these I was

able to run each property's location against the officially demarcated police district and elementary school geographies, which were obtained through the use of publicly available shape files - this also facilitated map-making for the purpose of analysis and presentation. In this way I was able to assign each property to its constituent police district and elementary school, after which merging was trivial. This collection of variables formed the regression equation like so:

$$\widehat{market.value} = b_0 + b_1 * ln(Total\ LivableArea) + b_2 * Exterior\ Condition + b_3 * Interior\ Condition + b_4 * View\ Type + b_5 * City\ Hall\ Distance + b_6 * Type\ 1\ Crime + b_7 * Type\ 2\ Crime + b_8 * Elementary\ School + b_9 * School\ Performance\ Score$$

Where b_0 refers to the intercept of the regression model. The equation above is not an exact representation of the actual regression equation as elementary school and view type are categorical variables. As such, each of their constituent levels are represented as dummy variables, taking the value of 0 or 1 depending on whether or not they are present in a given observation. Since each of the 160 schools is essentially an independent variable with its own coefficient, including the full equation is not practical. R's 'Caret' package was used to construct the multiple regression model.

In addition to the multiple regression model, a random forest model was also used. I selected random forest for two principal reasons; first, the large quantity of data and relatively high number of weak to moderately strong predictors suggested that an ensemble learning method may be successful, and second, given the complexity of the task and relatively large number of independent variables, assumption violations in the multiple regression model seemed all but certain. As a non-parametric method that makes no assumptions about the data, random forest is a robust alternative to linear regression. This flexibility comes at the cost of interpretability and computational time. Several implementations of the random forest algorithm exist within R, I used the implementation found in the 'ranger' package for its speed relative to its peers. Note that the categorical variables were not included in the random forest model. To assess model overfitting and out of sample error, an 80/20 train-test split was done on the data - model specific results come from the 'train' data, and accuracy results come from the 'test' data. All residuals were calculated by exponentiating the logged predictions and subtracting them from the original market values.

Results

The summary output for the multiple regression model are below:

Figure 4: Linear Model Summary

	Dependent variable:
	.outcome
Observations	362,780
R^2	0.894
Adjusted R^2	0.894
Residual Std. Error	0.260 (df = 362606)
F Statistic	17,739.920*** (df = 173; 362606)
Note:	*p<0.1;* ***p<0.05;*** p<0.01

The model coefficients can be found in Figure 5 in the appendix of this paper. From the results of the linear model, the model adjusted R^2 is 89.41%; which means that 89.4% of the variation in market values was explained by the independent variables included in the model. The F-Statistic shows that at least 1 of the 173

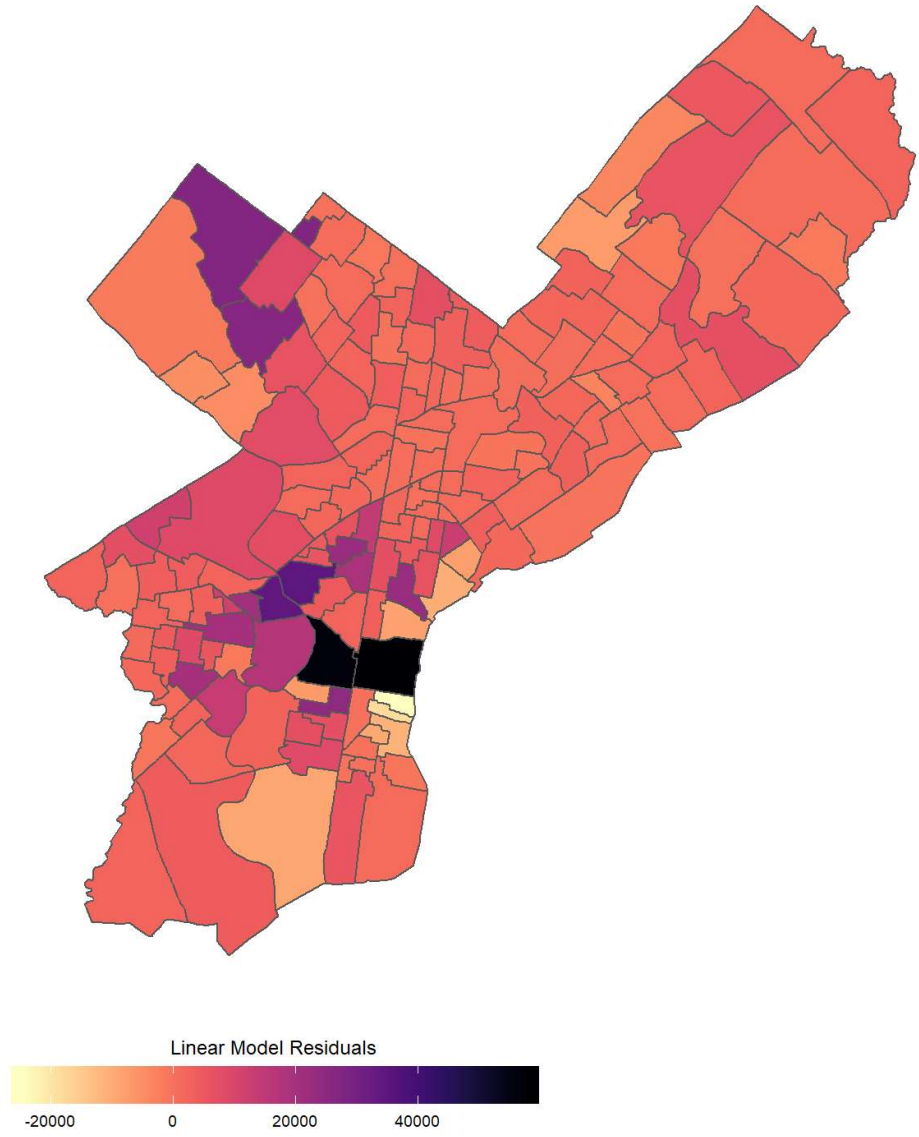
variables (including the factor levels) was statistically significant, and inspecting the individual variables shows that the majority of them attained significance as well. The random forest model has an R^2 of 95.82%, higher than the linear model. Ultimately, the more successful model is whichever one has the most prediction accuracy. As can be seen from the results below, the random forest model is significantly more accurate than the multiple regression model:

Figure 6: Error Metrics For Regression Models

Model	RMSE	MAE
Linear Model	87383	31166
Random Forest	46257	13741

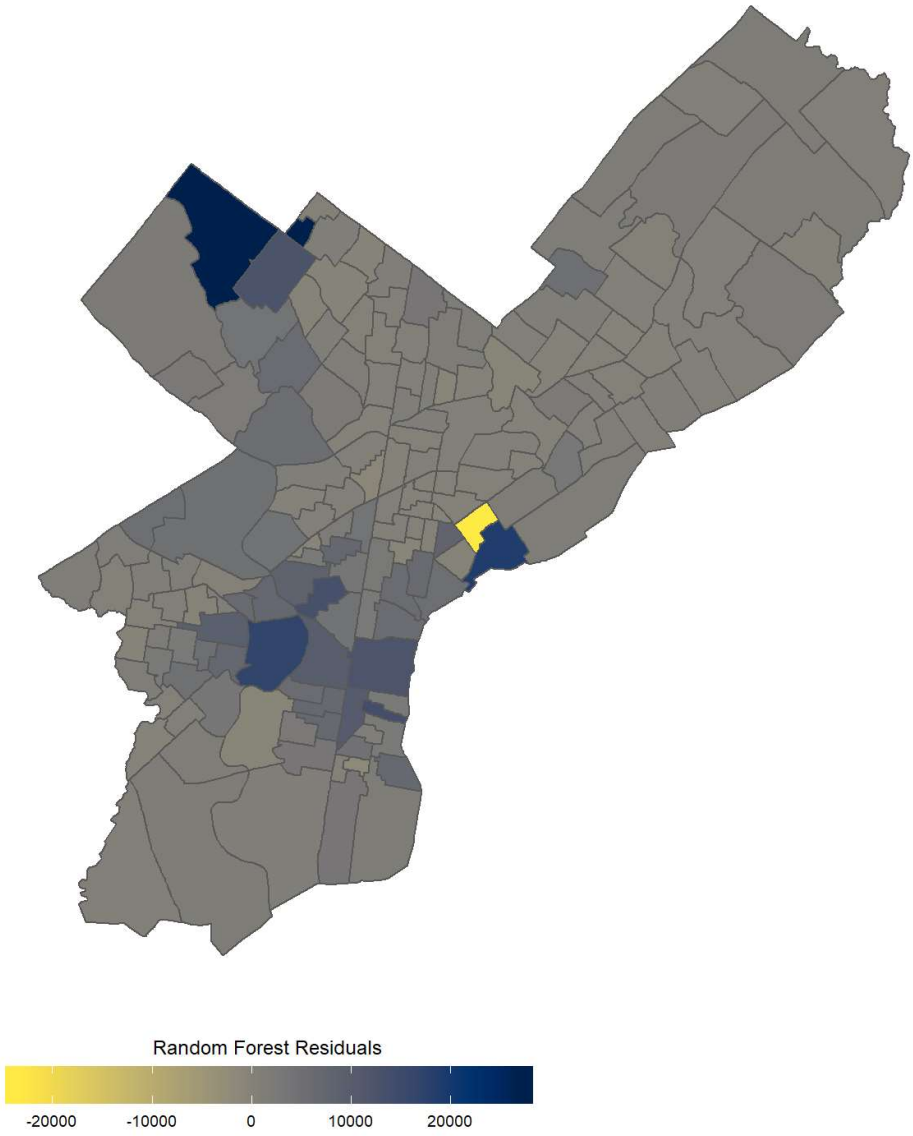
The extent of this difference in accuracy is best visualized through geography.

Figure 7: Geographic Dispersion of Linear Model Prediction Errors
Average Residual by Elementary School Catchment



The linear model clearly shows more prediction error in areas with higher property values, indicative of heteroscedasticity.

Figure 8: Geographic Dispersion of Random Forest Prediction Errors
Average Residual by Elementary School Catchment

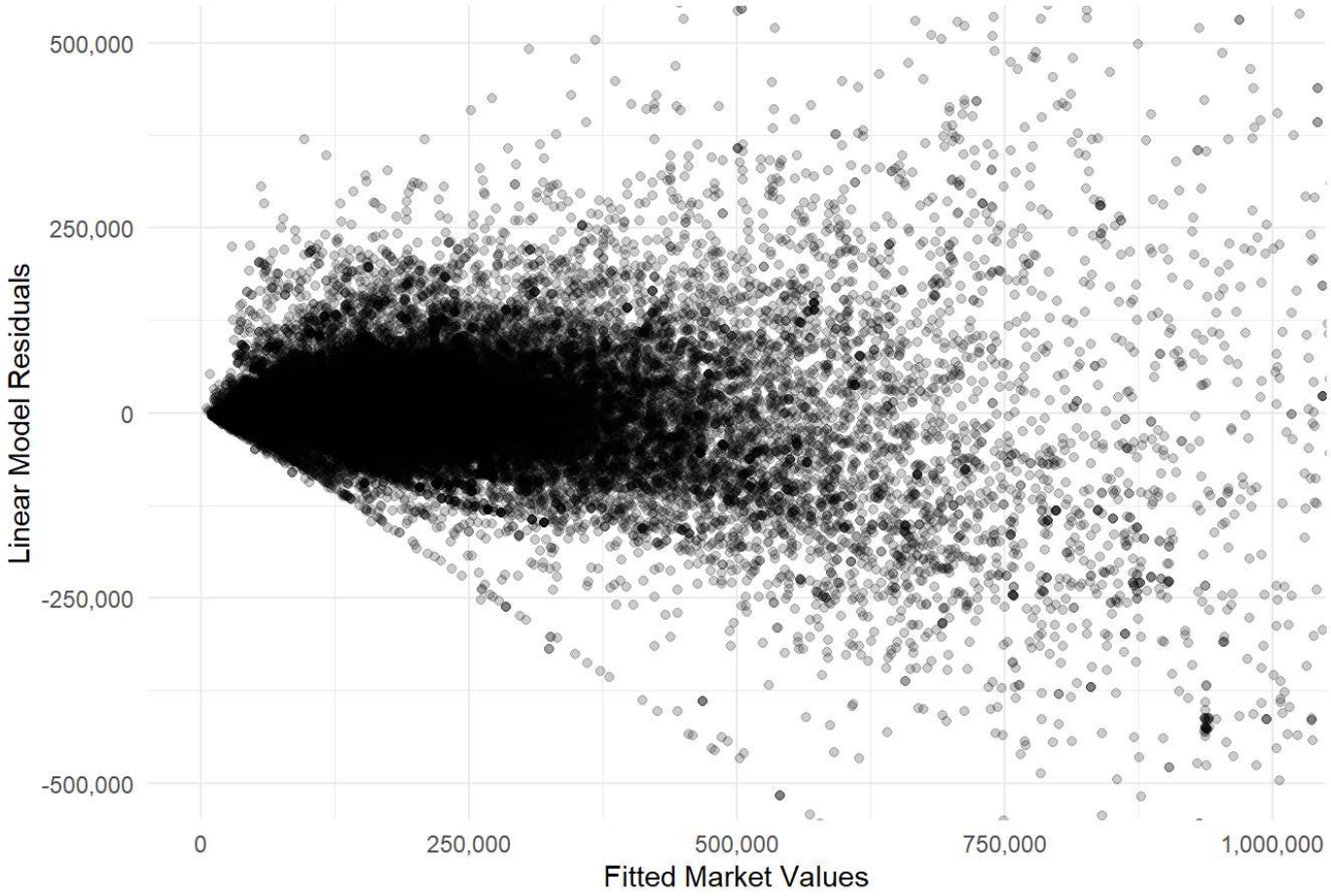


A visual scan shows that the magnitude of prediction errors is far smaller in the random forest model. In general, prediction errors are still the highest in areas of high market values, but to a much lower extent. It is thus fair to conclude that the random forest model is superior for this use case. Both models provide a measure of variable importance but unfortunately, they are difficult to assess and compare against one another due to the independent variables not being homogeneous between them. A truncated linear model variable importance is shown in the appendix in Figure 9 - it shows that total livable area and the elementary schools are the most important variables, based on their standardized T-values. This differs from the random forest variable importance, where Type 2 crime and the distance to city hall were the two most important variables, as measured by impurity. The lack of a uniform scale between these measures compounds the difficulty of comparing and interpreting them.

Limitations

Assumption violations within the linear model present a significant limitation to interpreting variable importance. Similar to the residual map, the residual plot below shows clear heteroscedasticity, which leads to bias within standard errors and by extension, test statistics.

Figure 11: Residuals vs Fitted Values



Multicollinearity is also present within the linear model. As such, the linear model's variable importance measure must be treated with skepticism, along with some of the regression coefficients. Interior and exterior condition are the most affected with a Pearson correlation of 98.4%, and exterior condition's small coefficient and lack of statistical significance may be a direct consequence of this. The linear model's assessment of total livable area and the elementary schools having the largest importance seems reasonable, but errors in the test statistics make the metric used to reach that conclusion tenuous at best. This may potentially be exacerbated by the way the crime variables were added into the data. As mentioned in the 'method' section, the crime variables are a daily weighted average of crime data spanning from January 2014 to November 2019. In reality, crime statistics are rarely presented in absolute terms, rather they are presented as a per capita rate. Calculating per capita rates proved extremely difficult because all population data sources I found were at the census tract or zip code level, whereas the crime data is segmented by police district. In order to calculate a per capita rate, it is necessary to link these disparate geographies - to either find the population of each police district, or the count of crimes for each census tract. Ultimately, it simply was not practical to pursue this further given time constraints and other obligations. Had I more time, I would have attempted a GIS solution - computing the area of one polygon (census tract) that lies within another (police district) would have enabled me to make an estimate. Future analysis would be improved by performing this calculation. The magnitudes of the crime variables as they stand are misrepresented. In absolute terms, the type 1 and type 2 crime rates of Center City are comparable to those of the impoverished areas in North and West Philadelphia, but this ignores the differing population densities of these areas. Center City has a population density of some 93,000 people per square mile, orders of magnitude higher than the density in the high crime areas of North and West Philadelphia, which, not coincidentally, have densities lower than the rest of the city due to depopulation. I believe this inaccuracy greatly misrepresents the importance of the crime variables, adding another source of doubt to the overall variable importance calculations. Continuing on the theme of geography, another limitation

comes from the lack of interaction between geographies in the linear model. The spatial effect of one elementary school catchment on another is not incorporated in the model. Empirically, it is easy to hypothesize that a given neighborhood that borders an impoverished area will have lower market values then an otherwise equivalent neighborhood that borders a wealthy area. Future analysis would again be improved by quantifying this effect.

As stated in the 'method' section, the random forest model did not include either of the two categorical variables. This is partly because the ranger implementation of random forest can only handle a certain number of levels within categorical variables. Since the elementary school variable has 160 levels, to include it requires converting the data into a sparse matrix. Unfortunately, technical constraints made this unfeasible. The run-time of the random forest model with school catchment and view type included spanned several hours. It would have most likely produced a richer, more accurate model, yet the computational cost was too high to justify. No hyper-parameter tuning was attempted for the same reason. Had I more time, I would have included all variables in both models. This would enable easier comparison between the variable importance metrics between the two.

Finally, another important limitation is that of the data itself. As mentioned in the 'data' section of this paper, I uncovered numerous inconsistencies and errors within the data during the process of data cleaning. For all of these that I uncovered and accounted for, there are likely many more that I did not notice, potentially increasing the error with the models. Additionally, potentially useful variables like the number of bedrooms and bathrooms were discarded because of the high proportion of missing values that they contained. One potential way to mitigate this would be to use a technique such as KNN-imputation to estimate the values of the missing data points. Assigning every missing observation the rounded average of the number of bedrooms/bathrooms found in homes in the same tranche of market values would potentially provide a good, if flawed, estimate. A greater concern may be found within the market values themselves. The properties dataset contains the officially assessed market values of every home in the city, and property tax calculations are based off of these assessments. Unsurprisingly, there has been substantial criticism directed towards the methodology that the city used to arrive at these figures. If the response variable itself was flawed, then by extension the entire analysis was as well. An interesting exercise would be to compare these valuations to those found on a realty website such as Zillow and see the difference. The lack of insight into the data-generation process presents a potential limitation over all of the independent variables from the properties dataset - I have no knowledge on how these were recorded. If the methodology to record these values was incorrect, then so too would conclusions derived from their effects on market value. Ultimately, lack of insight into the true data-generation process is very common - this is often the constraint within which the real world operates.

Conclusion

In this paper, I attempted to predict property values in Philadelphia and use the results to assess the factors that contribute to the vast discrepancy between market values in various parts of the city. While that goal itself has not been fully realized, I believe this analysis has provided a solid foundation which can be expanded and improved upon in pursuit of that goal. I have found that the non-parametric random forest has far superior performance for this purpose when compared to multiple linear regression, indicating that other non-parametric regression techniques such as gradient boosting may be better alternatives to test against the random forest. The linear model indicated that total livable area and the elementary school catchment were the most important variables, whereas in the random forest type 2 crime and the distance to city hall were the most important. Including all variables in the random forest would provide a more accurate basis of comparison between the

models in regards to variable importance. Most of all, further analysis is needed to validate these results, especially in light of the assumption violations in the multiple regression model. I hope that the results of this analysis can inform and improve further analysis, which I hope that I myself will have the chance to undertake.

References

Philadelphia, Pa Metropolitan Division: Nonfarm Employment and Labor Force Data https://www.bls.gov/regions/mid-atlantic/data/xg-tables/ro3fx9524.htm (https://www.bls.gov/regions/mid-atlantic/data/xg-tables/ro3fx9524.htm)

Philadelphia - 2018 Safety and Justice Challenge Fact Sheet. (n.d.). Retrieved December 20, 2019, from http://www.safetyandjusticechallenge.org/wp-content/uploads/2018/10/Philadelphia-Safety-Justice-Challenge-Fact-Sheet.pdf (http://www.safetyandjusticechallenge.org/wp-content/uploads/2018/10/Philadelphia-Safety-Justice-Challenge-Fact-Sheet.pdf).

Philadelphia 2019 - The State of the City. (n.d.). Retrieved December 20, 2019, from https://www.pewtrusts.org/en/research-and-analysis/reports/2019/04/11/philadelphia-2019 (https://www.pewtrusts.org/en/research-and-analysis/reports/2019/04/11/philadelphia-2019).

Appendix - Project Code

Main Project Script

```
# Explore the Philadelphia Properties Dataset
# Predict home values with regression techniques
# Load packages
library(data.table) # Fast data manipulation
library(tidyverse) # plotting, functional programming, factor and date functions
library(caret) # modeling
library(sf) # GIS tools
# 1 - Data Cleaning ------------------------------------------------------------------------
-------
# Load
# Properties contains the housing data, crime contains the crime data, school contains school
catchment
properties_SF.Trim <- fread('Econometrics Project\\opa_properties_23.10.19.csv')
crime_dt <- fread('Econometrics Project\\arrest_data_daily_by_district_csv.csv')
school_dt <- fread('Econometrics Project\\School.Data.Trimmed.csv')


# Focusing on Properties first
# Filter the data
properties_SF.Trim <- properties_SF.Trim[category_code_description == 'Single Family']


# Class of the variables
variable_class <- properties_SF.Trim[, lapply(.SD, class)]
variable_class <- variable_class %>%
  gather(key = 'Variable', value = 'Class')

# Most columns are characters. Many categorical data columns are listed as integers, not fact
ors
# Many extraneous columns - will remove
Delete_Cols <- variable_class$Variable[c(2, 3, 4, 10, 11, 12, 14, 15, 18, 19, 22,
                                         24, 25, 26, 29, 30, 31, 32, 33, 34,
                                         36, 41, 42, 44, 46, 47, 48, 49, 50, 51, 52,
                                         53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63,
                                         66, 67, 68, 69)]
# Remove The Columns
properties_SF.Trim <- properties_SF.Trim[, !..Delete_Cols]

# Crimes Data
# Classify crimes and aggregate by police district
# Using FBI's UCR Standard
crime_dt[, Type1.Crimes := Homicide + Rape + `Robbery/Gun` + `Robbery/Other` +
           `Aggravated Assault/Gun` + `Aggravated Assault/Other` + `Burglary/Residential` +
           `Burglary/Commercial` + `Theft of Motor Vehicle Tag` + `Theft from Person` +
           `Theft from Auto` + `Retail Theft` + Theft + `Auto Theft`][,
            Type2.Crimes := `Drug Possession` + `Drug Sales` + DUI + `All Other Offenses`][,
            'date_formatted' := as.Date(date_value,
            format = '%m/%d/%Y')][, 'Year.Formatted' := lubridate::year(date_formatted)]

# Load the PPD Shapefile
# Will use this to assign properties to their Police District
PPD.Districts <- read_sf('Econometrics Project\\PPD.Boundaries')
```

```
PPD.Districts <- st_as_sf(PPD.Districts)
PPD.Districts <- st_transform(PPD.Districts, crs = 4326)

# Two districts have since been merged with others - accounting for that here
crime_dt <- crime_dt[, dc_district := as_factor(dc_district)][,
           dc_district := fct_collapse(dc_district, `3` = '4', `22` = '23')][
           !(dc_district %in% c(71))]

# Aggregate and Summarise by Year and District
crimeSummary <- crime_dt[, .(Total.Crimes = sum(Type1.Crimes, Type2.Crimes),
                Total.Type1 = sum(Type1.Crimes), Total.Type2 = sum(Type2.Crimes)),
                by = .(Year.Formatted, dc_district)]

# Compute Crime Rates
crimeSummary.Melt <- crimeSummary %>%
  pivot_wider(names_from = Year.Formatted, values_from = c(Total.Crimes, Total.Type1, Total.T
ype2))
crimeSummary.Melt <- setDT(crimeSummary.Melt)
crimeDailyRates <- crimeSummary.Melt[, .(Police.District = dc_district,
                                         `2014Type1.Rate` = Total.Type1_2014 / 365,
                                         `2015Type1.Rate` = Total.Type1_2015 / 365,
                                         `2016Type1.Rate` = Total.Type1_2016 / 366,
                                         `2017Type1.Rate` = Total.Type1_2017 / 365,
                                         `2018Type1.Rate` = Total.Type1_2018 / 365,
                                         `2019Type1.Rate` = Total.Type1_2019 / 296,
                                         `2014Type2.Rate` = Total.Type2_2014 / 365,
                                         `2015Type2.Rate` = Total.Type2_2015 / 365,
                                         `2016Type2.Rate` = Total.Type2_2016 / 366,
                                         `2017Type2.Rate` = Total.Type2_2017 / 365,
                                         `2018Type2.Rate` = Total.Type2_2018 / 365,
                                         `2019Type2.Rate` = Total.Type2_2019 / 296)
                                     ][, c('Type1.WeightedAvg', 'Type2.WeightedAvg') :=
                                         .(0.05 * `2014Type1.Rate` + 0.1 * `2015Type1.Rate` +
                                           0.15 * `2016Type1.Rate` + 0.2 * `2017Type1.Rate` +
                                           0.225 * `2018Type1.Rate` + 0.275 * `2019Type1.Rate
`,
                                           0.05 * `2014Type2.Rate` + 0.1 * `2015Type2.Rate` +
                                           0.15 * `2016Type2.Rate` + 0.2 * `2017Type2.Rate` +
                                           0.225 * `2018Type2.Rate` + 0.275 * `2019Type2.Rate
`
                                           )
                                     ]

# Reduce to just the variables going into the properties dataset
crimeDailyRates <- crimeDailyRates[, .(Police.District, Type1.WeightedAvg, Type2.WeightedAv
g)]

# School Data
ES.Catchment.Area <- read_sf('Econometrics Project\\Catchment_ES_2017-18')

# ST_as_SF ensures that the Catchment is an SF object
# Will enable merging later
```

```
ES.Catchment.Area <- st_as_sf(ES.Catchment.Area, crs = 4326)
ES.Catchment.Area <- st_transform(ES.Catchment.Area, crs = 4326)

# Trim the School Performance Data
# This includes more than just public schools, but all that's needed are the performance metr
ics
school_dt <- school_dt[, c(3, 22)]


# 2 - EDA ----------------------------------------------------------------------------------
-----
# Distribution of Market Values is Strongly Skewed to the Right
# Taking the log may normalize the data
properties_SF.Trim[, Log.MktVal := log(market_value)]


# Taking the log resulted in some -Inf's. Inspection revealed that these observations had mar
ket values of 0
# Will remove them here
Zero.MktVal <- properties_SF.Trim[Log.MktVal == -Inf]

# There are 69 properties here with market values of 0
# Will remove these
properties_SF.Trim <- properties_SF.Trim[!is.infinite(properties_SF.Trim$Log.MktVal)]

# Removing market value outliers - these were clearly mistakes
properties_SF.Trim <- properties_SF.Trim[market_value < 100000000][
  !is.na(market_value)
]

# Number of rooms
# Almost none of these align with the number of bathrooms + number of bedrooms
# Unclear how a 'room' is defined here
properties_SF.Trim[, number_of_rooms := NULL]

# Now depth
# Will remove this variable
properties_SF.Trim[, depth := NULL]

# Garage Spaces
# This variable misrepresents garage spaces for condo's
# Will list a condo as having 30 spaces, as though it were a 30 car garage
properties_SF.Trim[, garage_spaces := NULL]
properties_SF.Trim[, garage_type := NULL]

# Number of Bathrooms and Bedrooms
# Both have 100K observations with 0 bedrooms or bathrooms
properties_SF.Trim <- properties_SF.Trim[!(location %in% c('900 S 12TH ST', '812 S 13TH ST'
))]
properties_SF.Trim[, c('number_of_bathrooms', 'number_of_bedrooms') := NULL]

# Number of Stories
```

```
# 100K observations with 0 stories
properties_SF.Trim[, number_stories := NULL]

# Basements
# 1/3rd missing values, will remove
properties_SF.Trim[, basements := NULL]

# Total Area
# Some properties that have a listed area of zero
# Isolating on those reveals that many observations have total area of zero but non zero tota
l livable area
# Will remove total area
properties_SF.Trim[, total_area := NULL]

# Total Livable Area
# 526 observations that show as having an area of zero
# Will remove these
# Some properties with very high total livable areas are not single family homes
# Many are housing for religious societies (Convents)
# Others are clearly mistakes
properties_SF.Trim <- properties_SF.Trim[total_livable_area > 0]
properties_SF.Trim <- properties_SF.Trim[
  !owner_1 %in% c('REV DENNIS J DOUGHERTY', 'DOUGHERTY DENNIS J', 'PHILADELPHIA UNIVERSITY',
                  'DENNIS J DOUGHERTY   ', 'TALMUDICAL YESHIVA OF PHI', "SAINT JOSEPH'S UNIVE
RSITY",
                  'SAINT JOSEPHS UNIVERSITY', 'CHURCH CHRISTIAN COMPASSI', 'CHRIST EMPOWERMEN
T TEMPLE',
                  'ST JAMES CATHOLIC', 'CARMELITE CONVENT OF', '    1439-61 NORTH 31ST STREE
T',
                  'UNITY MISSION CHURCH', 'KWAN UM SA BUDDHIST', 'SECOND BAPTIST CHURCH',
                  'UNITED COMMUNITIES', 'SOMERSET LLC', 'INTERCOMMUNITY ACTION INC',
                  'REV DENNIS DOUGHERTY', 'INDONESIA BETHEL CHURCH B', 'DARE TO IMAGINE CHURC
H IN',
                  'FRIENDS BEHAVORIAL HEALTH', 'CHOICE ACADEMICS INC', 'FRIENDS REHABILITATIO
N PR',
                  'ARCHBISHOP OF PHILADELPHI', 'HOLMESBURG BAPTIST CHURCH', 'DOMINICAN FATHER
S & BROTH',
                  'TALMUDICAL YESHIVA OF PHI', 'JAMESON EVANGELISTIC', '1439-61 NORTH 31ST ST
REET')]


# Removing Philadelphia Housing Authority, Redevelopment Authority, and Land Bank houses
properties_SF.Trim <- properties_SF.Trim[!(owner_1 %in%
                                     c('PHILADELPHIA HOUSING AUTH', 'PHILA HOUSING AUTHORIT
Y',
                                     'PHILADELPHIA LAND BANK', 'PHILADELPHIA HOUSING',
                                     'PHILADELPHIA REDEVELOPMEN', 'PHILA REDEVELOPMENT AUT
H',
                                     'PHILA HOUSING AUTH', 'PHILA REDEVELOPMENT'))]

# Removing Condo Parking Spaces and Residential Air Rights
properties_SF.Trim <- properties_SF.Trim[
```

```
  !(building_code_description) %in% c('CONDO PARKING SPACE', 'AIR RIGHTS RESIDENTIAL')]

# Removing NA's and 0's in Exterior and Interior Condition
properties_SF.Trim <- properties_SF.Trim[
  !is.na(exterior_condition)][
  !is.na(interior_condition)][!(interior_condition %in% 0)][
  !(exterior_condition %in% 0)]

# Removing Observations with Year Built of '0'
properties_SF.Trim <- properties_SF.Trim[!(year_built) %in% c('0')]

# Removing Observations with Missing Lat/Long
properties_SF.Trim <- properties_SF.Trim[!is.na(lng)]

# Most of the remaining non ID variables will go in the model

# 3 - Transformation ------------------------------------------------------------------------
----
# Recode Categorical Vars as factors/numeric where applicable
# Exterior Condition
properties_SF.Trim[, exterior_condition := as_factor(exterior_condition)]
properties_SF.Trim[, exterior_condition := fct_recode(exterior_condition,
                                    '7' = '1', '6' = '2', '5' = '3',
                                    '4' = '4', '3' = '5',
                                    '2' = '6', '1' = '7')]
properties_SF.Trim[,
    exterior_condition := as.numeric(as.character(properties_SF.Trim$exterior_condition))]

# Interior Condition
properties_SF.Trim[, interior_condition := as_factor(interior_condition)][
  , interior_condition := fct_recode(interior_condition,
                                    '7' = '1', '6' = '2', '5' = '3',
                                    '4' = '4', '3' = '5',
                                    '2' = '6', '1' = '7')]
properties_SF.Trim[,
    interior_condition := as.numeric(as.character(properties_SF.Trim$interior_condition))]

# View Type
properties_SF.Trim[, c('view_type') :=
                     lapply(.SD, as_factor), .SDcols = c('view_type')]

# Add the distance variable - Using City Hall as a proxy for 'downtown' distance
Properties.Long.Lat <- properties_SF.Trim[, .(objectid, lng, lat)]
# Compute Distance - Using the Haversine Method
properties_SF.Trim[,
  City.Hall.Distance := geosphere::distHaversine(Properties.Long.Lat[, c(2:3)],
                                                 c(-75.1635112, 39.952335), r = 3963.1905919
)]

# To add in Police District and School Catchment, I will create an SF object with the Long.La
t DT above
Properties.Long.Lat <- st_as_sf(Properties.Long.Lat, coords = c('lng', 'lat'), crs = 4326)
```

```
# Apply Point in Polygon Algorithm to See which Police District each address belongs to
PD.Intersection <- st_intersection(Properties.Long.Lat, PPD.Districts)
st_geometry(PD.Intersection) <- NULL
PD.Intersection <- PD.Intersection[, c(1, 5)]

# Merge
properties_SF.Trim <- merge(properties_SF.Trim, PD.Intersection, by = 'objectid')

# Rename and set as factor
setnames(properties_SF.Trim, old = 'DISTRICT_', new = 'Police.District')
properties_SF.Trim[, Police.District := as_factor(Police.District)]

# Add in the Crime Data
properties_SF.Trim <- merge(properties_SF.Trim, crimeDailyRates, by = 'Police.District')

# Repeating For Elementary Schools
ES.Intersection <- st_intersection(Properties.Long.Lat, ES.Catchment.Area)
st_geometry(ES.Intersection) <- NULL
ES.Intersection <- ES.Intersection[, c(1, 2, 4)]

# Merge with Intersection DT
properties_SF.Trim <- merge(properties_SF.Trim, ES.Intersection, by = 'objectid')

# Merge with Performance Data
# Repeating the ULCS code to make merging easier
colnames(school_dt) <- c('ES_ID', 'School.Score')
school_dt[, ES_ID := as.character(school_dt$ES_ID)]
properties_SF.Trim <- merge(properties_SF.Trim, school_dt, by = 'ES_ID')
properties_SF.Trim[, School.Score := as.numeric(properties_SF.Trim$School.Score)]


# 4 - Model ------------------------------------------------------------------------
# Running Descriptive Stats
# Measures of Centrality
MktVal.Centrality.Final <- data.table(
  Mean = mean(properties_SF.Trim$market_value, na.rm = TRUE),
  `Trimmed Mean 5%` = mean(properties_SF.Trim$market_value, na.rm = TRUE, trim = 0.05),
  `Trimmed Mean 10%` = mean(properties_SF.Trim$market_value, na.rm = TRUE, trim = 0.1),
  Median = median(properties_SF.Trim$market_value, na.rm = TRUE)
)
Log.MktVal.Centrality.Final <- data.table(
  Mean = mean(properties_SF.Trim$Log.MktVal, na.rm = TRUE),
  `Trimmed Mean 5%` = mean(properties_SF.Trim$Log.MktVal, na.rm = TRUE, trim = 0.05),
  `Trimmed Mean 10%` = mean(properties_SF.Trim$Log.MktVal, na.rm = TRUE, trim = 0.1),
  Median = median(properties_SF.Trim$Log.MktVal, na.rm = TRUE)
)
# Exponentiate back the log tranformed centrality measures
Exp.Centrality.Final <- map_df(Log.MktVal.Centrality.Final, exp)

# Combine Results
Centrality.DT.Final <- bind_rows(
```

```
  MktVal.Centrality.Final,
  Log.MktVal.Centrality.Final,
  Exp.Centrality.Final
) %>%
  map_df(~round(., 2)) %>%
  mutate(Distribution = c('Market Values', 'Log Transformed Market Values', 'Exponentiated Lo
g Values')) %>%
  select(5, 1:4)

# Measures of variability
MktVal.Variability.Final <- data.table(
  `Standard Deviation` = sd(properties_SF.Trim$market_value, na.rm = TRUE),
  `Median Absolute Deviation` = mad(properties_SF.Trim$market_value, na.rm = TRUE),
  `Interquartile Range` = IQR(properties_SF.Trim$market_value, na.rm = TRUE),
  Range = range(properties_SF.Trim$market_value, na.rm = TRUE)[2] - range(properties_SF.Trim
$market_value, na.rm = TRUE)[1]
) %>% map_df(~round(., 2))

# Split Data into Test and Train
properties.Model <- properties_SF.Trim[
  , .(objectid, exterior_condition, interior_condition, total_livable_area, year_built, view_
type,
      Log.MktVal, City.Hall.Distance, Type1.WeightedAvg, Type2.WeightedAvg, ES_Short, School.
Score)][
  , total_livable_area := log(total_livable_area)
      ]

in.train <- createDataPartition(properties.Model$year_built, p = 0.8, list = FALSE, times = 1
)

properties.Train <- properties.Model[in.train]
properties.Test <- properties.Model[-in.train]

# Extracting the original market values and object ID's to merge back and calculate residuals
Market.Value.DT <- properties_SF.Trim[, .(objectid, market_value)]

# Test correlations
Numeric.Correlation <- data.frame(Correlation = sapply(select_if(properties.Model, is.numeri
c),
       cor, y = properties.Model$Log.MktVal, use = 'complete.obs'))

# Linear Model
set.seed(11202019)
LM.Params <- trainControl(method = "boot", number = 25)
Linear.Model <- train(
  x = properties.Train[, c(2, 3, 4, 6, 8, 9, 10, 11, 12)],
  y = properties.Train$Log.MktVal,
  method = 'lm',
  trControl = LM.Params
)

# Add in Residuals
```

```
properties.Train <- properties.Train %>%
  modelr::add_predictions(Linear.Model, var = 'Log.LM.Predictions') %>%
  left_join(y = Market.Value.DT, by = 'objectid') %>%
  mutate(Scaled.LM.Predictions = exp(Log.LM.Predictions),
         LM.Residuals = market_value - Scaled.LM.Predictions)


# Random Forest
RF.Model <- ranger::ranger(Log.MktVal ~ total_livable_area + exterior_condition + interior_co
ndition +
                           City.Hall.Distance + Type1.WeightedAvg + Type2.WeightedAvg + Schoo
l.Score,
                           data = properties.Train,
                           num.trees = 500, mtry = 4)


# Add predictions and residuals
setDT(properties.Train)
properties.Train[, c('Log.RF.Predictions') :=
                   RF.Model$predictions,][,
  c('Scaled.RF.Predictions') :=
      .(exp(Log.RF.Predictions))][,
  RF.Residuals := market_value - Scaled.RF.Predictions
  ]


# Consolidate Training Results
Training.Preds <- data.table(
  objectid = properties.Train$objectid,
  Scaled.LM.Predictions = properties.Train$Scaled.LM.Predictions,
  Sclaed.RF.Predictions = properties.Train$Scaled.RF.Predictions
)


Training.Preds <- merge(Training.Preds, Market.Value.DT, by = 'objectid')

Training.Accuracy <- data.frame(
  RMSE = map_dbl(Training.Preds[, c(2:3)], ~ RMSE(., Training.Preds$market_value)),
  MAE = map_dbl(Training.Preds[, c(2:3)], ~ MAE(., Training.Preds$market_value))
) %>%
  map_dfc(~round(.)) %>%
  mutate(
    Model = c('Linear Model', 'Random Forest')
  ) %>%
  select(3, 1, 2)


# Consolidate and calculate residuals for Test Accuracy
# Add in Market Value Test DT
# Test Predictions
properties.Test <- merge(properties.Test, Market.Value.DT, by = 'objectid')
properties.Test[,
```

```
 ':=' ('Log.LM.Predictions' = predict(Linear.Model, properties.Test),
      'Log.RF.Predictions' = predict(RF.Model, data = properties.Test)$predictions)][,
      c('Scaled.LM.Predictions', 'Scaled.RF.Predictions')
      := lapply(.SD, exp),
      .SDcols = c('Log.LM.Predictions', 'Log.RF.Predictions')][,
      c('LM.Resid', 'RF.Resid') := market_value - .SD,
    .SDcols = c('Scaled.LM.Predictions', 'Scaled.RF.Predictions')]


# Calculate Accuracy
Test.Accuracy <- data.frame(
  RMSE = map_dbl(properties.Test[, c(16, 17)], ~ RMSE(., properties.Test$market_value)),
  MAE = map_dbl(properties.Test[, c(16, 17)], ~ MAE(., properties.Test$market_value))
) %>%
  map_dfc(~round(.)) %>%
  mutate(Model = c(
    'Linear Model', 'Random Forest')) %>%
  select(3, 1:2)
```

Figure 1: Market Value Map

```
properties.ES.Avg <- properties_SF.Trim[, .(Mean = mean(market_value)), by = ES_ID]
# Merge
ES.Catchment.Area.Plot <- ES.Catchment.Area %>%
  left_join(y = properties.ES.Avg, by = 'ES_ID')
# Plot
ES.MktVal <- ggplot(ES.Catchment.Area.Plot) +
  geom_sf(aes(fill = Mean)) +
  coord_sf(expand = TRUE) +
  ggthemes::theme_map() +
  viridis::scale_fill_viridis(name = 'Mean Market Value',
                              guide = guide_colorbar(
                                direction = 'horizontal',
                                barwidth = unit(85, units = 'mm'),
                                title.position = 'top',
                                title.hjust = 0.5
                              ), labels = scales::comma, option = 'magma', direction = 1) +
  theme(legend.position = 'bottom') +
  labs(title = 'Property Values in Philadelphia',
       subtitle = 'Average Market Value by Elementary School Catchment Area, 2019',
       caption = 'Source: Philadelphia Office of Property Assesments')
ES.MktVal
```

Figure 2: Distribution of Market Values

```
Mkt.Val.Dist <- ggplot(properties_SF.Trim) +
  geom_histogram(aes(market_value), binwidth = 2 * IQR(properties_SF.Trim$market_value, na.rm
= TRUE) / length(properties_SF.Trim$market_value)^(1/3)) +
  coord_cartesian(xlim = c(0, 1000000)) +
  scale_x_continuous(breaks = seq(0, 1000000, by = 100000), labels = scales::comma) +
  labs(x = 'Property Value', y = 'Count', title = 'Figure 2: Distribution of Home Values in P
hiladelphia', subtitle = 'Single Family Homes Only',
       caption = 'Source: Philadelphia Office of Property Assessment') +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 50, hjust = 1))
Mkt.Val.Dist
```

Figure 3: Log-Transformed Distribution of Market Values

```
Log.Dist <- ggplot(properties_SF.Trim) +
  geom_histogram(aes(Log.MktVal), binwidth = 2 * IQR(properties_SF.Trim$Log.MktVal, na.rm = T
RUE) / length(properties_SF.Trim$Log.MktVal)^(1/3)) +
  coord_cartesian(xlim = c(6, 18)) +
  labs(x = 'Natural Log of Property Values', y = 'Count',
       title = 'Figure 3: Log Transformed Distribution of Philadelphia Property Values',
       subtitle = 'Single Family Homes Only',
       caption = 'Source: Philadelphia Office of Property Assessment') +
  theme_minimal()
Log.Dist
```

Figure 4: Linear Model Summary

```
stargazer::stargazer(Linear.Model$finalModel, title = 'Figure 4: Linear Model Summary', keep
 = 0)
```

Figure 5: Linear Model Variable Results

```
knitr::kable(broom::tidy(Linear.Model$finalModel), caption = 'Linear Model Results')
```

Linear Model Results

term	estimate	std.error	statistic	p.value
(Intercept)	5.7569673	0.0128316	448.6561420	0.0000000
exterior_condition	0.0014579	0.0035416	0.4116399	0.6806036
interior_condition	0.2442040	0.0035174	69.4269839	0.0000000
total_livable_area	0.7833639	0.0014949	524.0178891	0.0000000
view_type	-0.2671186	0.0102869	-25.9667532	0.0000000
view_typeC	0.1218644	0.0036468	33.4167864	0.0000000
view_typeH	-0.0020714	0.0059776	-0.3465354	0.7289406

term	estimate	std.error	statistic	p.value
view_type0	-0.2320896	0.0075521	-30.7319922	0.0000000
view_typeB	-0.1053905	0.0093746	-11.2420800	0.0000000
view_typeE	0.0036102	0.0069524	0.5192687	0.6035737
view_typeA	-0.0559779	0.0029824	-18.7696101	0.0000000
view_typeD	-0.0232204	0.0064162	-3.6190157	0.0002958
City.Hall.Distance	0.0231622	0.0015570	14.8764161	0.0000000
Type1.WeightedAvg	-0.0383282	0.0037887	-10.1165091	0.0000000
Type2.WeightedAvg	-0.0018381	0.0004959	-3.7066278	0.0002101
ES_ShortAllen, Ethan	-0.5497541	0.0115241	-47.7047996	0.0000000
ES_ShortAllen, Ethel	-1.9738295	0.0077167	-255.7878752	0.0000000
ES_ShortAmerican Paradigm CS (Birney)	-1.0729696	0.0102418	-104.7632950	0.0000000
ES_ShortAnderson	-1.1710380	0.0081977	-142.8495725	0.0000000
ES_ShortArthur	0.2300451	0.0075387	30.5151104	0.0000000
ES_ShortBache/Martin	0.2720186	0.0071287	38.1584921	0.0000000
ES_ShortBarry	-1.2876120	0.0075581	-170.3616138	0.0000000
ES_ShortBarton	-1.3410280	0.0090220	-148.6394359	0.0000000
ES_ShortBelmont CS	-1.3618714	0.0114308	-119.1405790	0.0000000
ES_ShortBethune	-1.7391119	0.0091310	-190.4617796	0.0000000
ES_ShortBlaine	-1.7597794	0.0101629	-173.1580483	0.0000000
ES_ShortBlankenburg	-1.5305981	0.0093336	-163.9882894	0.0000000
ES_ShortBregy	-0.0091843	0.0070353	-1.3054548	0.1917388
ES_ShortBridesburg	-0.3757153	0.0092554	-40.5941532	0.0000000
ES_ShortBrown, HA	-0.5332164	0.0085245	-62.5512123	0.0000000
ES_ShortBrown, JH	-0.4741656	0.0131232	-36.1319610	0.0000000
ES_ShortBryant	-1.2866205	0.0081352	-158.1555820	0.0000000
ES_ShortCarnell	-0.8495097	0.0098957	-85.8467700	0.0000000
ES_ShortCassidy	-0.7577098	0.0081780	-92.6527134	0.0000000
ES_ShortCatharine	-1.1293081	0.0085465	-132.1370320	0.0000000
ES_ShortCayuga	-1.5853870	0.0100873	-157.1662027	0.0000000

term	estimate	std.error	statistic	p.value
ES_ShortChilds	-0.3939887	0.0068874	-57.2040759	0.0000000
ES_ShortComegys	-1.2556780	0.0074624	-168.2681696	0.0000000
ES_ShortComly	-0.4386210	0.0198540	-22.0922864	0.0000000
ES_ShortCook-Wiss	-0.2985306	0.0088441	-33.7546692	0.0000000
ES_ShortCooke	-1.1274440	0.0106374	-105.9885238	0.0000000
ES_ShortCramp	-1.7626532	0.0094425	-186.6721068	0.0000000
ES_ShortCrossan	-0.3636025	0.0126111	-28.8318533	0.0000000
ES_ShortDay	-0.6247703	0.0108416	-57.6271160	0.0000000
ES_ShortdeBurgos	-1.9229580	0.0109812	-175.1136799	0.0000000
ES_ShortDecatur	-0.4641809	0.0197806	-23.4665287	0.0000000
ES_ShortDick	-1.8029289	0.0097759	-184.4257329	0.0000000
ES_ShortDisston	-0.6872833	0.0121224	-56.6952802	0.0000000
ES_ShortDobson	-0.3251924	0.0095990	-33.8777953	0.0000000
ES_ShortDuckrey	-1.6903047	0.0075892	-222.7262281	0.0000000
ES_ShortDunbar	-0.3433568	0.0109516	-31.3521347	0.0000000
ES_ShortEdmonds, FS	-0.6399814	0.0113947	-56.1649952	0.0000000
ES_ShortElkin	-1.9583094	0.0092983	-210.6085069	0.0000000
ES_ShortEllwood	-0.7349290	0.0118536	-62.0004769	0.0000000
ES_ShortEmlen	-0.8439324	0.0103541	-81.5069969	0.0000000
ES_ShortFarrell	-0.3319174	0.0140333	-23.6521155	0.0000000
ES_ShortFell	-0.1160715	0.0068986	-16.8252462	0.0000000
ES_ShortFinletter	-0.6469189	0.0103300	-62.6250047	0.0000000
ES_ShortFitzpatrick	-0.4205871	0.0180080	-23.3556257	0.0000000
ES_ShortForrest	-0.6818020	0.0128118	-53.2167290	0.0000000
ES_ShortFox Chase	-0.2407684	0.0132622	-18.1544946	0.0000000
ES_ShortFrank	-0.4608920	0.0160072	-28.7927143	0.0000000
ES_ShortFranklin ES	-0.8492638	0.0103249	-82.2535563	0.0000000
ES_ShortGideon	-1.4678820	0.0128436	-114.2891039	0.0000000
ES_ShortGirard	-0.4155975	0.0063435	-65.5155449	0.0000000

term	estimate	std.error	statistic	p.value
ES_ShortGlobal Leadership CS (Huey)	-0.9185901	0.0079104	-116.1250270	0.0000000
ES_ShortGompers	-0.7682281	0.0093476	-82.1841507	0.0000000
ES_ShortGreenberg	-0.2823639	0.0160495	-17.5933102	0.0000000
ES_ShortGreenfield	0.6826097	0.0062937	108.4587944	0.0000000
ES_ShortHackett	-0.2140978	0.0067742	-31.6048073	0.0000000
ES_ShortHamilton	-1.1991066	0.0094189	-127.3087323	0.0000000
ES_ShortHancock-LaBrum	-0.4237085	0.0173453	-24.4278400	0.0000000
ES_ShortHarrington	-1.0544437	0.0073467	-143.5254937	0.0000000
ES_ShortHartranft	-1.8338089	0.0104352	-175.7336765	0.0000000
ES_ShortHenry	-0.3094741	0.0108414	-28.5455946	0.0000000
ES_ShortHeston	-1.4528196	0.0089467	-162.3869078	0.0000000
ES_ShortHolme	-0.4557053	0.0150219	-30.3361212	0.0000000
ES_ShortHopkinson	-0.8471014	0.0115016	-73.6504580	0.0000000
ES_ShortHouston	-0.3199341	0.0109085	-29.3287784	0.0000000
ES_ShortHowe	-0.8514280	0.0106710	-79.7889861	0.0000000
ES_ShortHunter	-1.4150216	0.0110920	-127.5711996	0.0000000
ES_ShortJackson	0.2432028	0.0069437	35.0248747	0.0000000
ES_ShortJenks, Abram	-0.0224364	0.0086081	-2.6064329	0.0091494
ES_ShortJenks, John	-0.0046869	0.0122715	-0.3819369	0.7025084
ES_ShortJuniata Park	-0.5662366	0.0109644	-51.6430671	0.0000000
ES_ShortKearny	0.0738050	0.0086538	8.5286271	0.0000000
ES_ShortKelley, WD	-1.2527779	0.0094755	-132.2124425	0.0000000
ES_ShortKelly, JB	-0.9948424	0.0080956	-122.8868552	0.0000000
ES_ShortKenderton	-1.8154294	0.0094773	-191.5558479	0.0000000
ES_ShortKey	-0.2213468	0.0079646	-27.7913701	0.0000000
ES_ShortKirkbride	-0.0371970	0.0080148	-4.6410224	0.0000035
ES_ShortLamberton ES	-0.5238121	0.0091353	-57.3395297	0.0000000
ES_ShortLawton	-0.7545423	0.0116192	-64.9391603	0.0000000
ES_ShortLea	-0.2211749	0.0113585	-19.4721933	0.0000000

term	estimate	std.error	statistic	p.value
ES_ShortLingelbach	-0.5693471	0.0103761	-54.8711679	0.0000000
ES_ShortLocke	-0.9308860	0.0085484	-108.8960740	0.0000000
ES_ShortLoesche	-0.4645478	0.0186623	-24.8922609	0.0000000
ES_ShortLogan	-1.2204223	0.0097462	-125.2205332	0.0000000
ES_ShortLongstreth	-1.3170243	0.0079449	-165.7689781	0.0000000
ES_ShortLowell	-0.8126229	0.0092117	-88.2167382	0.0000000
ES_ShortLudlow	-0.1605264	0.0080741	-19.8817031	0.0000000
ES_ShortMarshall, J	-1.1778113	0.0117484	-100.2529409	0.0000000
ES_ShortMarshall, T	-1.0445906	0.0110342	-94.6686930	0.0000000
ES_ShortMastery CS (Cleveland)	-1.6686274	0.0081387	-205.0246494	0.0000000
ES_ShortMastery CS (Clymer)	-1.8810805	0.0092881	-202.5263495	0.0000000
ES_ShortMastery CS (Douglass)	-1.0820831	0.0096884	-111.6889070	0.0000000
ES_ShortMastery CS (Harrity)	-1.1206802	0.0073101	-153.3052930	0.0000000
ES_ShortMastery CS (Mann)	-0.8341937	0.0074980	-111.2548997	0.0000000
ES_ShortMastery CS (Pastorius)	-1.2861801	0.0097966	-131.2879722	0.0000000
ES_ShortMastery CS (Smedley)	-1.1292930	0.0111297	-101.4665204	0.0000000
ES_ShortMastery CS (Wister)	-1.3756325	0.0097124	-141.6374370	0.0000000
ES_ShortMayfair	-0.3532151	0.0125615	-28.1188056	0.0000000
ES_ShortMcCall	0.5571353	0.0062381	89.3117150	0.0000000
ES_ShortMcCloskey	-0.6616994	0.0132499	-49.9400476	0.0000000
ES_ShortMcClure	-1.5452713	0.0090902	-169.9925593	0.0000000
ES_ShortMcDaniel	-0.7595695	0.0066118	-114.8800773	0.0000000
ES_ShortMcKinley	-1.1369245	0.0116385	-97.6861812	0.0000000
ES_ShortMcMichael	-0.9031902	0.0100008	-90.3113723	0.0000000
ES_ShortMeade	-0.6940721	0.0109229	-63.5429091	0.0000000
ES_ShortMeredith	0.5202693	0.0078550	66.2339687	0.0000000
ES_ShortMifflin	-0.1880591	0.0081094	-23.1903130	0.0000000
ES_ShortMitchell	-1.3231241	0.0076199	-173.6395346	0.0000000
ES_ShortMoffet	-0.3431371	0.0081065	-42.3284231	0.0000000

term	estimate	std.error	statistic	p.value
ES_ShortMoore	-0.5942685	0.0108543	-54.7493716	0.0000000
ES_ShortMorris	-0.1870792	0.0067800	-27.5928588	0.0000000
ES_ShortMorrison	-1.1319219	0.0100292	-112.8622016	0.0000000
ES_ShortMorton	-1.0771592	0.0071935	-149.7402071	0.0000000
ES_ShortMunoz Marin	-1.7297829	0.0108270	-159.7650016	0.0000000
ES_ShortNebinger	0.3420889	0.0090327	37.8724784	0.0000000
ES_ShortOlney ES	-0.9645241	0.0098494	-97.9271944	0.0000000
ES_ShortOverbrook ES	-0.6170635	0.0099272	-62.1586566	0.0000000
ES_ShortPatterson	-0.9007869	0.0073956	-121.8008380	0.0000000
ES_ShortPeirce, TM	-1.7568090	0.0073162	-240.1273717	0.0000000
ES_ShortPenn Alexander	0.1477929	0.0107927	13.6937761	0.0000000
ES_ShortPennell	-0.9844915	0.0100607	-97.8556292	0.0000000
ES_ShortPennypacker	-0.7791217	0.0111340	-69.9767968	0.0000000
ES_ShortPenrose	-0.6549517	0.0093022	-70.4086205	0.0000000
ES_ShortPhila Arts CS (HR Edmunds)	-0.7766798	0.0107869	-72.0022028	0.0000000
ES_ShortPollock	-0.4058145	0.0143560	-28.2679495	0.0000000
ES_ShortPotter-Thomas	-1.8224056	0.0094806	-192.2244011	0.0000000
ES_ShortPowel	-0.1835854	0.0135697	-13.5290515	0.0000000
ES_ShortPrince Hall	-0.9122144	0.0104581	-87.2253714	0.0000000
ES_ShortRhawnhurst	-0.3285407	0.0127615	-25.7446125	0.0000000
ES_ShortRhoads, J	-1.4947063	0.0078800	-189.6847261	0.0000000
ES_ShortRhodes, EW	-1.5590104	0.0075137	-207.4898248	0.0000000
ES_ShortRichmond	-0.4664835	0.0085436	-54.6006224	0.0000000
ES_ShortRoosevelt ES	-1.1030884	0.0102857	-107.2444669	0.0000000
ES_ShortRowen	-0.7783802	0.0107786	-72.2150018	0.0000000
ES_ShortSharswood	-0.1563923	0.0074955	-20.8649268	0.0000000
ES_ShortShawmont	-0.3334170	0.0107760	-30.9406926	0.0000000
ES_ShortSheppard	-1.9963951	0.0119717	-166.7597821	0.0000000
ES_ShortSheridan	-1.5350255	0.0091598	-167.5828482	0.0000000

term	estimate	std.error	statistic	p.value
ES_ShortSolis-Cohen	-0.5687644	0.0115935	-49.0589838	0.0000000
ES_ShortSouthwark	0.0325663	0.0070715	4.6052836	0.0000041
ES_ShortSpring Garden	-0.0058264	0.0109794	-0.5306610	0.5956540
ES_ShortSpruance	-0.6795479	0.0109993	-61.7807297	0.0000000
ES_ShortStanton, EM	-0.1289551	0.0070010	-18.4195747	0.0000000
ES_ShortStearne	-1.1810788	0.0110847	-106.5507746	0.0000000
ES_ShortSteel	-1.6028877	0.0084740	-189.1534455	0.0000000
ES_ShortSullivan	-0.8912900	0.0107581	-82.8483486	0.0000000
ES_ShortTaggart	-0.3343266	0.0080177	-41.6986473	0.0000000
ES_ShortTaylor	-1.6246027	0.0095406	-170.2827441	0.0000000
ES_ShortUniversal CS (Alcorn)	-0.8108779	0.0068193	-118.9085121	0.0000000
ES_ShortUniversal CS (Bluford)	-1.1133451	0.0070787	-157.2800368	0.0000000
ES_ShortUniversal CS (Creighton)	-0.8979352	0.0106082	-84.6456580	0.0000000
ES_ShortUniversal CS (Daroff)	-1.4953003	0.0079474	-188.1504811	0.0000000
ES_ShortVare-Washington	-0.0186907	0.0069299	-2.6971256	0.0069944
ES_ShortWaring	0.2150402	0.0076851	27.9815491	0.0000000
ES_ShortWashington, Martha	-1.1902111	0.0163448	-72.8189281	0.0000000
ES_ShortWebster	-1.1667247	0.0103129	-113.1328624	0.0000000
ES_ShortWelsh	-1.7176420	0.0117153	-146.6150734	0.0000000
ES_ShortWillard	-1.2823239	0.0101709	-126.0783145	0.0000000
ES_ShortWright	-1.9192313	0.0080472	-238.4975577	0.0000000
ES_ShortZiegler	-0.6934313	0.0129367	-53.6017135	0.0000000

Figure 6: Accuracy Table

```
Test.Accuracy <- data.frame(
  RMSE = map_dbl(properties.Test[, c(16, 17)], ~ RMSE(., properties.Test$market_value)),
  MAE = map_dbl(properties.Test[, c(16, 17)], ~ MAE(., properties.Test$market_value))
) %>%
  map_dfc(~round(.)) %>%
  mutate(Model = c(
    'Linear Model', 'Random Forest')) %>%
  select(3, 1:2)

knitr::kable(Test.Accuracy, caption = 'Figure 6: Error Metrics For Regression Models')
```

Figure 7: Linear Model Residuals Map

```
LM.resid <- properties.Test %>%
  group_by(ES_Short) %>%
  summarise(`Mean Residual` = mean(LM.Resid))
# Merge
ES.LM.Residuals <- ES.Catchment.Area %>%
  left_join(y = LM.resid, by = 'ES_Short')

LM.Residuals.Plot <- ggplot(ES.LM.Residuals) +
  geom_sf(aes(fill = `Mean Residual`)) +
  ggthemes::theme_map() +
  viridis::scale_fill_viridis(
    name = 'Linear Model Residuals',
    guide = guide_colorbar(
      direction = 'horizontal',
      barwidth = unit(85, units = 'mm'),
      title.position = 'top',
      title.hjust = 0.5
    ),
    option = 'magma', direction = -1) +
  theme(legend.position = 'bottom') +
  labs(title = 'Figure 7: Geographic Dispersion of Linear Model Prediction Errors',
       subtitle = 'Average Residual by Elementary School Catchment')
LM.Residuals.Plot
```

Figure 8: Random Forest Residuals Map

```
RF.Resid <- properties.Test %>%
  group_by(ES_Short) %>%
  summarise(`Mean Residual` = mean(RF.Resid))
# Merge
ES.RF.Residuals <- ES.Catchment.Area %>%
  left_join(y = RF.Resid, by = 'ES_Short')
# Plot
RF.Residuals.Plot <- ggplot(ES.RF.Residuals) +
  geom_sf(aes(fill = `Mean Residual`)) +
  ggthemes::theme_map() +
  viridis::scale_fill_viridis(
    name = 'Random Forest Residuals',
    guide = guide_colorbar(
      direction = 'horizontal',
      barwidth = unit(85, units = 'mm'),
      title.position = 'top',
      title.hjust = 0.5
    ),
    option = 'cividis', direction = -1) +
  theme(legend.position = 'bottom') +
  labs(title = 'Figure 8: Geographic Dispersion of Random Forest Prediction Errors',
       subtitle = 'Average Residual by Elementary School Catchment')
RF.Residuals.Plot
```

Figure 9: Linear Model Variable Importance

```
plot(varImp(Linear.Model), top = 20)
```

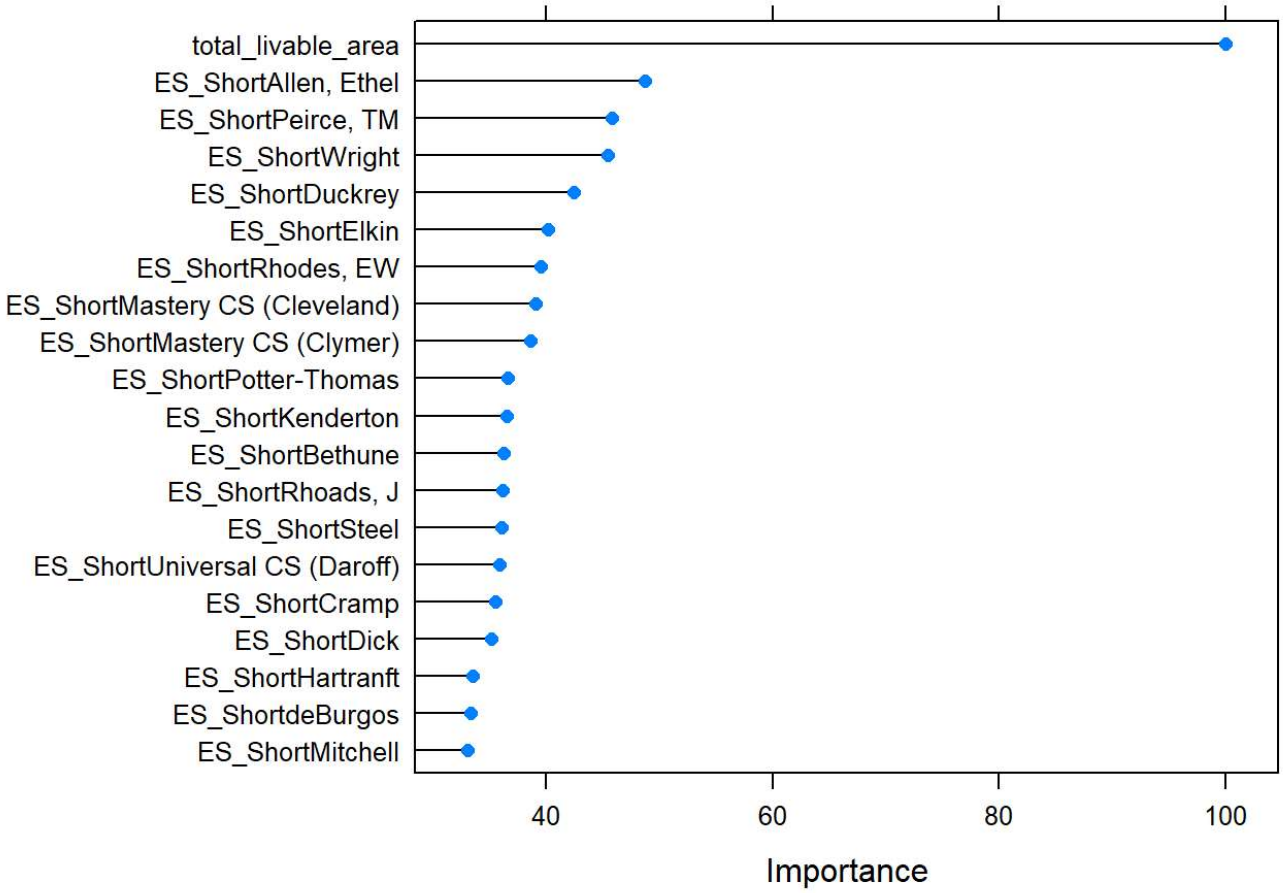


Figure 10: Random Forest Variable Importance

```
RF.Importance <- as.data.frame(ranger::importance(RF.Model))
RF.Importance <- setDT(RF.Importance, keep.rownames = TRUE)
colnames(RF.Importance) <- c('Variable', 'Importance Score')
knitr::kable(RF.Importance, caption = 'Figure 10: Random Forest Importance Scores')
```

Figure 10: Random Forest Importance Scores

Variable	Importance Score
total_livable_area	39156.75
exterior_condition	7195.62
interior_condition	15596.00
City.Hall.Distance	55337.49
Type1.WeightedAvg	18329.94
Type2.WeightedAvg	65192.61
School.Score	26840.42

Figure 11: Linear Model Residual vs Fitted Plot

```
Resid.Plot <- ggplot(data = properties.Test) +
  geom_point(aes(x = Scaled.LM.Predictions, y = LM.Resid), alpha = 0.2) +
  coord_cartesian(xlim = c(0, 1000000), ylim = c(-500000, 500000)) +
  scale_x_continuous(labels = scales::comma) +
  scale_y_continuous(labels = scales::comma) +
  labs(x = 'Fitted Market Values', y = 'Linear Model Residuals', title = 'Figure 11: Residual
s vs Fitted Values') +
  theme_minimal()
Resid.Plot
```