Seattle Housing Market Analysis

Justin Mai

Contents

1. Abstract

Keywords: King County Home Sales, Multiple Linear Regression, Prediction Modeling

2. Introduction

Within a highly populated like the Great Seattle and King County area, looking to be a first time homeowner or to purchase a home within the area in general is a tall task. Prices can fluctuate based on several factors making it difficult to make financial decisions. The goal of this report is to develop a model that can support home buyers, investors, and real estate agents navigate the housing market by including the variables that are the most impactful to home prices. Using historical home sales data in King County from 2023-2025, this report aims to use machine learning tools used in multiple linear regression to predict prices. Our goals are **(1) Discover the variables and factors that are significant to forecasting King County home sales using linear regression and ANOVA tools (2) See how well our model predicts prices on our test dataset**

3. Data

3.1 Data Overview

The primary dataset we are using was developed by *Andy Krause* who works as the Director of Valuation and Market Dynamics at Zillow. The dataset he produced was developed with the goal of creating an open access user tool to support in analyzing the housing market. Our dataset consists of 33,333 different observations which represents home sales from 2023 to the start of 2025. Our model will be primarily used to make predictions on `sale_price` and the dataset consists of continuous variables like `land_val`, `sqft`, and `sqft_lot`, and categorical variables like `city`, `zoning`, and `subdivision`. The dataset comes with 45+ predictors for us to use. See https://github.com/andykrause/kingCoData for more details.

3.2 Data Splitting

To avoid data leakage, the influence of training data on testing data, we are splitting our data before applying any type of manipulation. We will be using a standard 80% by 20% split between training and testing data respectively. The data manipulation and transformations will be applied to the training data before modeling, our optimal model developed by the training data will be used on the testing data.

```
sales <- sales %>%
  mutate(sale_date = as.Date(sale_date),
         across(where(is.character), as.factor))
```

After applying splitting our data, we have 26667 observations in the training data and 6666 observations in the testing data which are both considered sufficient amounts of data. We have chosen the split randomly to eliminate bias in splitting and are also using `set.seed(101)` for the experiment to be reproducible. (See appendix for details)

3.3 Data Manipulation

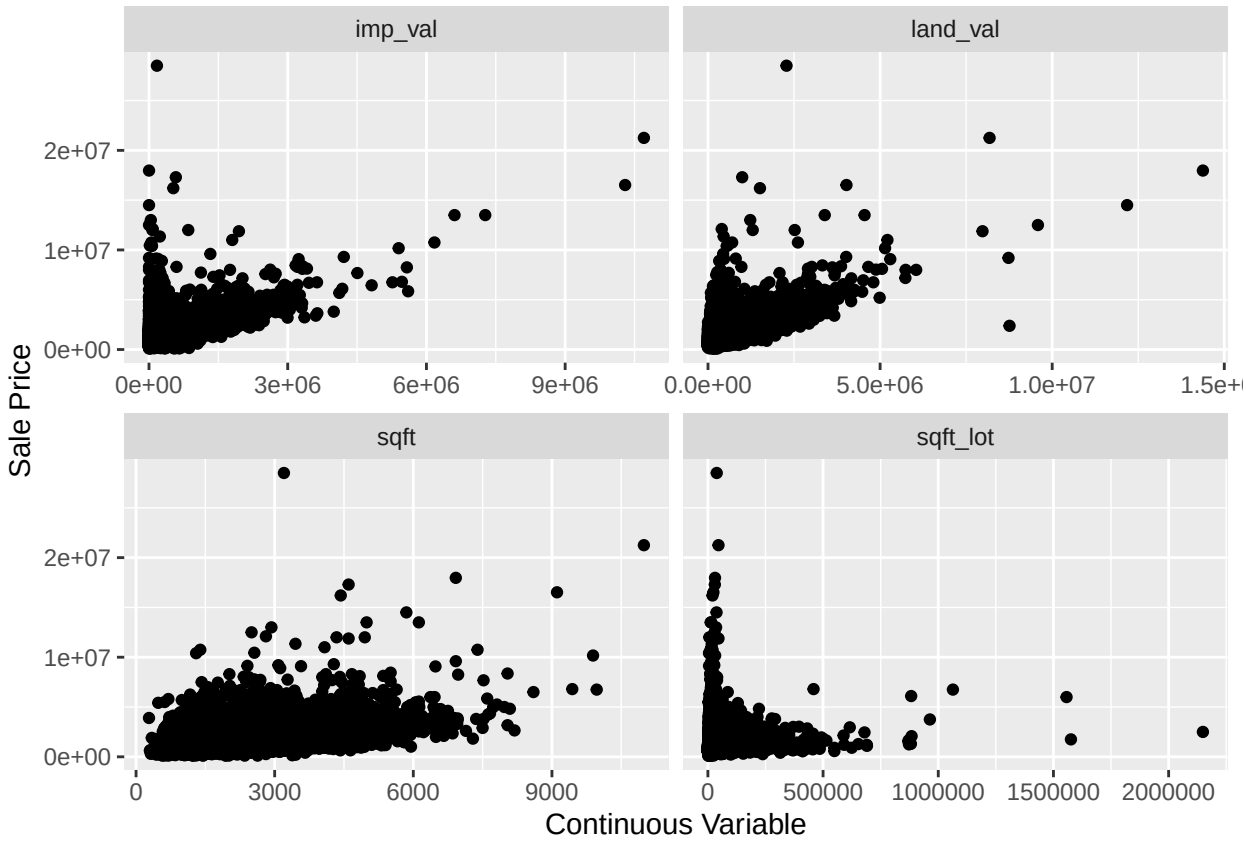
Within our data manipulation process, we started by filtering our data to be data from 2023-2025 because that is the scope of the sales data that we want to model off of. We then recognized that there were too many cities for our data to handle and based off our report for correlation, we found that cities within the same region rise in price at around the same rate. This enables us to create a `region` variable that represents multiple cities listed, giving us a more condensed model without forfeiting any values. We then removed variables that we deemed were unnecessary like `longitude` and `latitude`. We also saw from modeling with the full model that the `view` variables were mostly not significant at $\alpha = 0.05$ so we consolidated it into one binary variable that shows the value 1 if the home does have an exotic view or 0 if not. Before finalizing this model, we checked for NA values to see if there were any and if there were any patterns within NA values. We want the data to be an adequate representation of the population, so ensuring that NA values didn't strongly impact one category was important for us. Luckily we didn't identify any NULL values that resulted in bias within the dataset.

3.3 Data Transformation

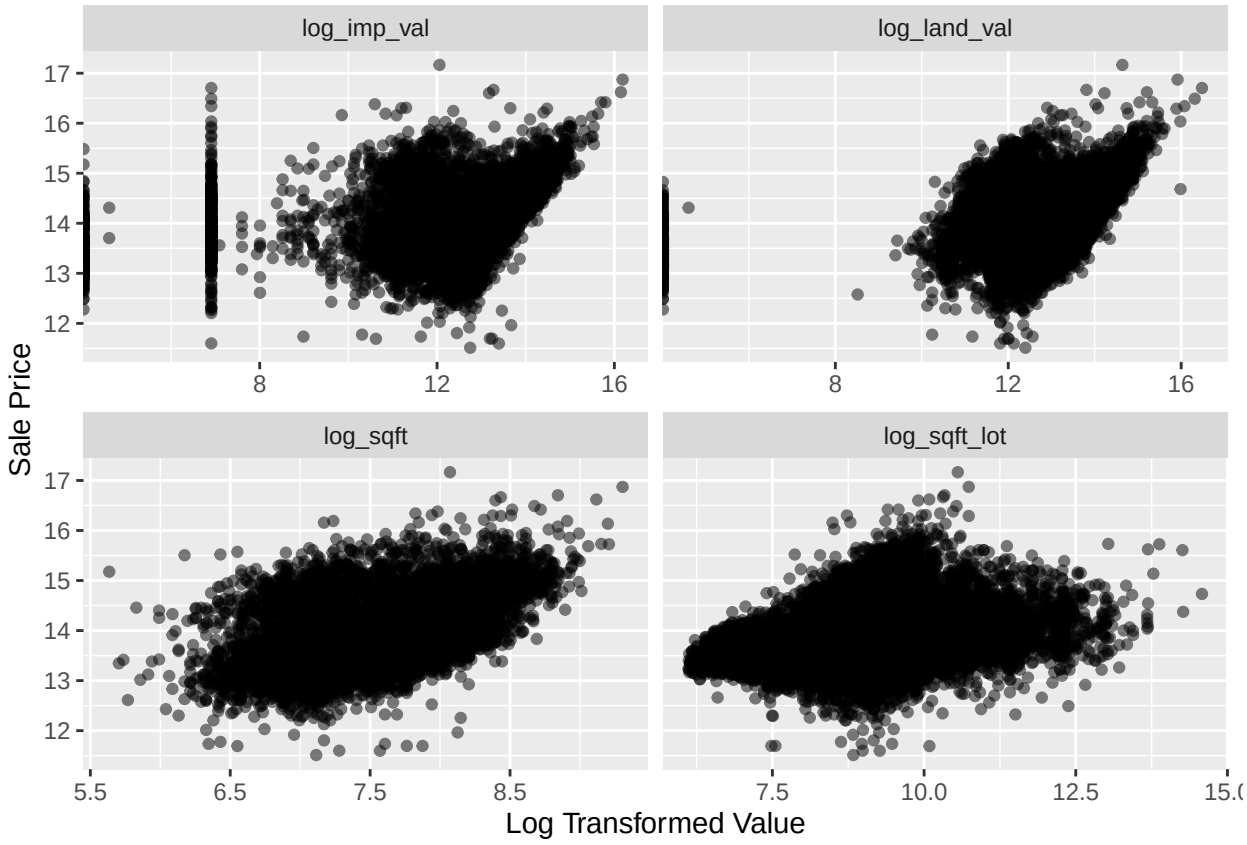
Linear Modeling Assumptions:

- **Linearity**: The relationship between the predictor and response is linear.
- **Independence**: All observations are independent of one another (pair-wise independence).
- **Homoscedasticity**: The variance of residuals is constant across predictor levels.
- **Residual Normality**: The residuals follow a normal distribution.

To start with our tests, we first checked residual normality among our continuous variables when plotting against our response variable of `sale_price`. We can see that the residuals weren't normal with each plot demonstrating left skewness with many high outliers.



To satisfy this test, we applied a log-transformation on our response and continuous variables to normalize the scatterplots.




```
summary(lm(sale_price ~ . , data = train))
```

```
##
## Call:
## lm(formula = sale_price ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2065057  -176919   -24135   105212 23805454
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  2.235e+06  4.730e+05   4.727 2.29e-06 ***
## sale_date                    1.511e+02  1.515e+01   9.976  < 2e-16 ***
## area                        -1.016e+03  1.700e+02  -5.976 2.32e-09 ***
## present_use                 -2.045e+03  6.538e+02  -3.128 0.001765 **
## land_val                     2.041e-01  1.110e-02  18.395  < 2e-16 ***
## imp_val                      5.506e-01  1.520e-02  36.228  < 2e-16 ***
## year_built                  -2.325e+03  1.861e+02 -12.495  < 2e-16 ***
## sqft_lot                     7.040e-01  8.635e-02   8.152 3.73e-16 ***
## sqft                         2.735e+02  1.291e+01  21.179  < 2e-16 ***
## sqft_1                      -2.515e+01  1.553e+01  -1.619 0.105501
## sqft_fbsmt                  -1.625e+02  1.825e+01  -8.905  < 2e-16 ***
## grade                        3.969e+04  5.416e+03   7.329 2.39e-13 ***
## fbsmt_grade                 -4.555e+03  2.026e+03  -2.248 0.024558 *
## condition                   -1.889e+04  4.907e+03  -3.850 0.000118 ***
## stories                     -1.004e+05  1.342e+04  -7.481 7.63e-14 ***
```

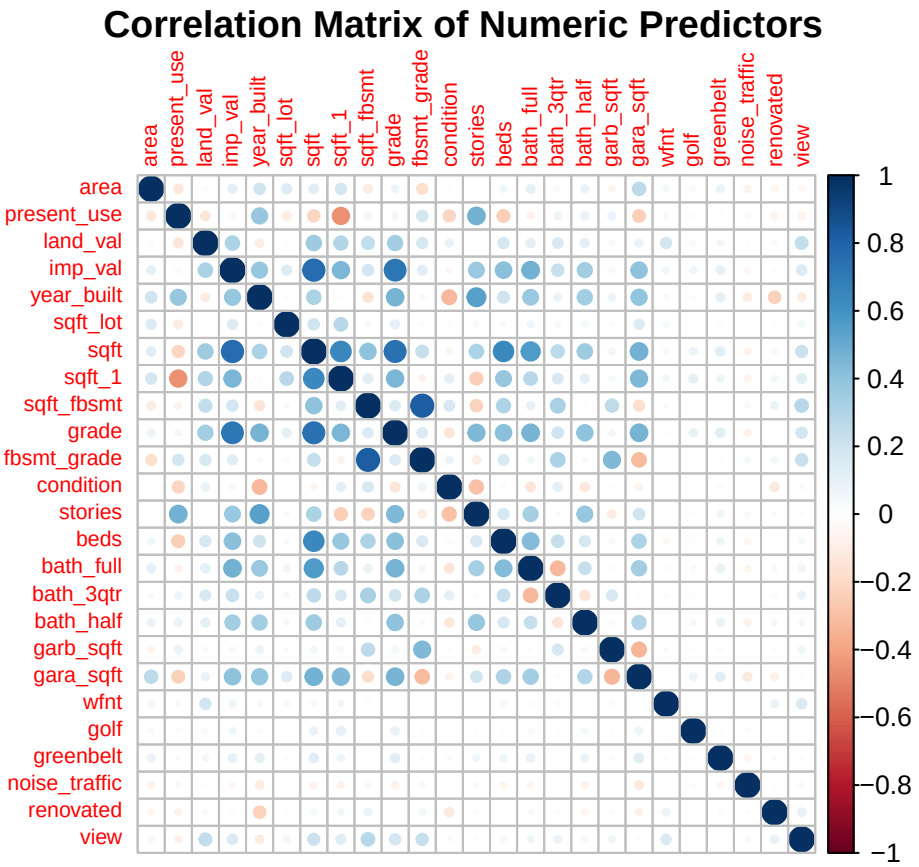
```
## beds                          -2.736e+04  4.896e+03  -5.588 2.31e-08 ***
## bath_full                     -2.879e+03  8.074e+03  -0.357 0.721371
## bath_3qtr                      1.523e+04  7.244e+03   2.102 0.035532 *
## bath_half                      2.320e+04  7.689e+03   3.018 0.002549 **
## garb_sqft                     -1.574e+02  2.287e+01  -6.885 5.91e-12 ***
## gara_sqft                     -1.089e+02  1.801e+01  -6.042 1.54e-09 ***
## wfnt                           1.502e+05  3.986e+03  37.677  < 2e-16 ***
## golf                           5.485e+04  4.103e+04   1.337 0.181324
## greenbelt                     -4.050e+04  2.000e+04  -2.025 0.042918 *
## noise_traffic                 -6.428e+04  5.829e+03 -11.027  < 2e-16 ***
## submarketB                     1.667e+05  2.098e+04   7.947 1.99e-15 ***
## submarketC                     2.356e+05  2.225e+04  10.589  < 2e-16 ***
## submarketD                     3.097e+05  2.190e+04  14.142  < 2e-16 ***
## submarketE                    -1.313e+04  2.574e+04  -0.510 0.610100
## submarketF                     2.588e+04  2.193e+04   1.180 0.237974
## submarketG                    -2.915e+05  3.819e+04  -7.631 2.40e-14 ***
## submarketH                    -3.826e+05  5.455e+04  -7.013 2.38e-12 ***
## submarketI                    -3.355e+05  3.756e+04  -8.932  < 2e-16 ***
## submarketJ                    -2.606e+05  3.306e+04  -7.884 3.31e-15 ***
## submarketK                    -2.426e+05  3.559e+04  -6.816 9.55e-12 ***
## submarketL                    -3.113e+05  4.094e+04  -7.604 2.96e-14 ***
## submarketM                    -3.509e+05  3.685e+04  -9.523  < 2e-16 ***
## submarketN                    -1.443e+05  3.750e+04  -3.849 0.000119 ***
## submarketO                     9.531e+04  3.755e+04   2.538 0.011142 *
## submarketP                     6.676e+04  3.743e+04   1.784 0.074496 .
## submarketQ                     2.497e+05  3.462e+04   7.214 5.59e-13 ***
## submarketR                     1.729e+05  3.582e+04   4.827 1.39e-06 ***
## submarketS                     1.880e+06  4.015e+04  46.813  < 2e-16 ***
## regionGeneral King County     -2.478e+05  2.086e+04 -11.876  < 2e-16 ***
## regionNorth King County       -3.689e+05  2.135e+04 -17.281  < 2e-16 ***
## regionPierce County           -1.998e+05  1.205e+05  -1.659 0.097211 .
## regionRural King County       -2.110e+05  2.890e+04  -7.300 2.96e-13 ***
## regionSeattle Area            -3.531e+05  3.490e+04 -10.116  < 2e-16 ***
## regionSouth King County       -2.287e+05  2.407e+04  -9.499  < 2e-16 ***
## regionSoutheast King County   -1.970e+05  2.129e+04  -9.251  < 2e-16 ***
## regionSouthwest King County   -3.022e+05  2.684e+04 -11.258  < 2e-16 ***
## regionUnknown                 -5.407e+05  3.166e+04 -17.081  < 2e-16 ***
## renovated                      5.717e+04  1.646e+04   3.472 0.000517 ***
## view                           3.071e+05  1.113e+04  27.583  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 512500 on 26335 degrees of freedom
##   (278 observations deleted due to missingness)
## Multiple R-squared:  0.6617, Adjusted R-squared:  0.661
## F-statistic: 971.8 on 53 and 26335 DF,  p-value: < 2.2e-16
```

4. Model

```r
num_vars <- train %>%
  select_if(is.numeric) %>%
  select(-sale_price)

corr_mat <- cor(num_vars, use = "pairwise.complete.obs")

corrplot(corr_mat, method = "circle", tl.cex = 0.7,
         title = "Correlation Matrix of Numeric Predictors",
         mar = c(0,0,1,0))
```



4.1 Model Assumptions / Diagnostics

4.2 Model Selection

5. Results

6. Discussion / Conclusion

7 Appendix

Data splitting process

```
set.seed(101)
n = nrow(sales)

test_indices <- sample(seq_len(n), size = floor(0.2 * n))

test <- sales[test_indices,]
train <- sales[-test_indices,]
```

Data manipulation process

```
train <- train %>%
  filter(sale_date >= "2023-01-01")

train <- train %>%
  mutate(region = case_when(
    city %in% c("SEATTLE", "SHORELINE", "LAKE FOREST PARK") ~ "Seattle Area",
    city %in% c("BELLEVUE", "REDMOND", "KIRKLAND", "MEDINA", "CLYDE HILL", "YARROW POINT", "HUNTS PO
    city %in% c("RENTON", "TUKWILA", "SEA-TAC", "DES MOINES", "BURIEN", "NORMANDY PARK") ~ "South Ki
    city %in% c("AUBURN", "FEDERAL WAY", "ALGONA", "PACIFIC", "KENT") ~ "Southwest King County",
    city %in% c("SAMMAMISH", "ISSAQUAH", "MAPLE VALLEY", "COVINGTON", "BLACK DIAMOND") ~ "Southeast
    city %in% c("WOODINVILLE", "KENMORE", "BOTHELL", "DUVALL") ~ "North King County",
    city %in% c("SNOQUALMIE", "NORTH BEND", "SKYKOMISH", "CARNATION", "ENUMCLAW") ~ "Rural King Coun
    city == "MILTON" ~ "Pierce County",
    city == "KING COUNTY" ~ "General King County",
    TRUE ~ "Unknown"
  )) %>%
  select(!c(sale_id,pinx,sale_nbr,sale_warning,join_status,join_year,latitude,longitude)) %>%
  mutate(renovated = ifelse(year_reno == 0, 0, 1)) %>%
  select(!c(city)) %>%
  mutate(view = ifelse(view_rainier >= 1 | view_olympics >= 1 | view_cascades >= 1 | view_territoria
  select(!c(view_rainier,view_olympics,view_cascades,view_territorial,view_skyline,view_sound,view_l
```

Pre-log transformation

```
train %>%
  pivot_longer(cols = c(sqft, sqft_lot, land_val, imp_val),
               names_to = "Variable", values_to = "Value") %>%
ggplot(aes(x = Value, y = sale_price)) +
  geom_point() +
  facet_wrap(~Variable, scales = "free_x") +
  labs(x = "Continuous Variable", y = "Sale Price")
```

Post-log transformation

```
train %>%
  mutate(across(c(sqft, sqft_lot, land_val, imp_val), log, .names = "log_{.col}")) %>%
  pivot_longer(cols = starts_with("log_"),
```

```
            names_to = "Variable", values_to = "Value") %>%
ggplot(aes(x = Value, y = log(sale_price))) +
geom_point(alpha = 0.5) +
facet_wrap(~Variable, scales = "free_x") +
labs(x = "Log Transformed Value", y = "Sale Price")
```