Final Project

Hunter Garrison, Kevin Smith, Reese Madsen, Giulio Martini

2024-04-26

Section 1: Introduction

In this project, we are working with a dataset comprising of the observed prices of houses and corresponding attributes of each house. This dataset is from the database website Kaggle and is comprised of 545 observations of houses in India with 13 variables recorded for each specific house. Below are the attributes of each house recorded in the dataset:

| Attribute	Description	Data Type
price	Price of the houses in rupees. (this is our response variable)	Integer
area	Area of a house in square feet	Integer
bedrooms	Number of bedrooms in house	Integer
bathrooms	Number of bathrooms in house	Integer
stories	Number of stories in house	Integer
mainroad	Whether or not house is connected to main road	Boolean
guestroom	Whether or not house has a guest room	Boolean
basement	Whether or not house has a basement	Boolean
hotwaterheating	Whether or not house has a hot-water heater	Boolean
airconditioning	Whether or not house has air conditioning	Boolean
parking	Number of parking spots at house	Integer
prefarea	Whether or not the house is in a preferred area	Boolean
furnishingstatus	Furnishing status of the house (furnished, semi-furnished, unfurnished)	String

The majority of our predictors in this dataset are of type boolean, with the remainders being integers and one categorical predictor comprising of strings. Below is the first 5 lines of our dataset (code 1):

price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement
13300000	7420	4	2	3	yes	no	no
12250000	8960	4	4	4	yes	no	no
12250000	9960	3	2	2	yes	no	yes
12215000	7500	4	2	2	yes	no	yes
11410000	7420	4	1	2	yes	yes	yes

| hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
no	yes	2	yes	furnished
no	yes	3	no	furnished
no	no	2	yes	semi-furnished
no	yes	3	yes	furnished
no	yes	2	no	furnished

We are attempting to create a linear regression model that best explains the variance in the price of the houses using the predictors in our dataset. In this report, we will explore our data by finding collinearity, outliers, influential points, and by checking error assumption violations. We will also discover which variables are the most important by using variable selection, and test OLS as well as GLS and WLS regressions. After doing this, we will combine all the knowledge we have gained about our data set in order to find the model in which we believe best explains the variance in the price of the houses.

We will now perform some exploratory data analysis to show the distribution of our response variable as well as some key relationships in this dataset. Below is a histogram of our response variable, price, as well as a table showing some key statistical information about the response (code 2):
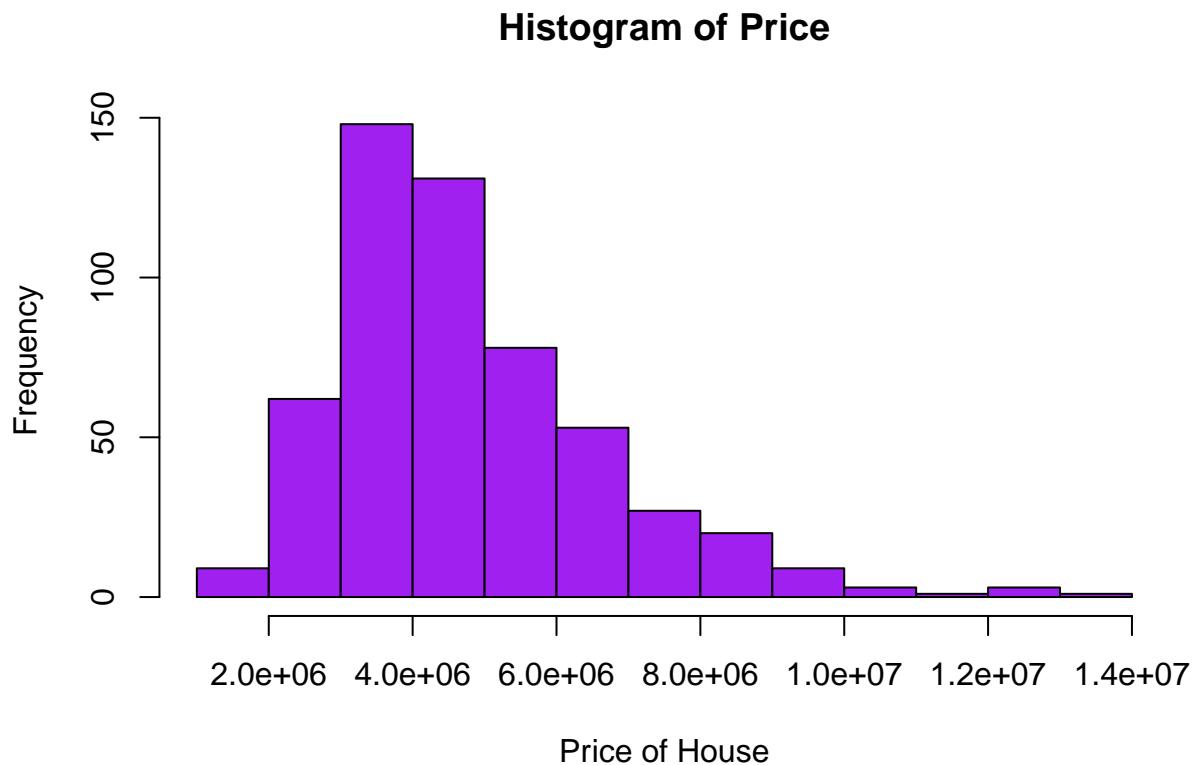
Histogram of Price


```
##       Min      Q1 Median    Mean      Q3      Max NA_Count
## 1 1750000 3430000 4340000 4766729 5740000 13300000        0
```

From our histogram and table, we see that our data is right skewed with a small amount of houses showing prices much farther than the mean. Below, we also show a scatter plot of price vs. area (code 3):

Price vs. Area
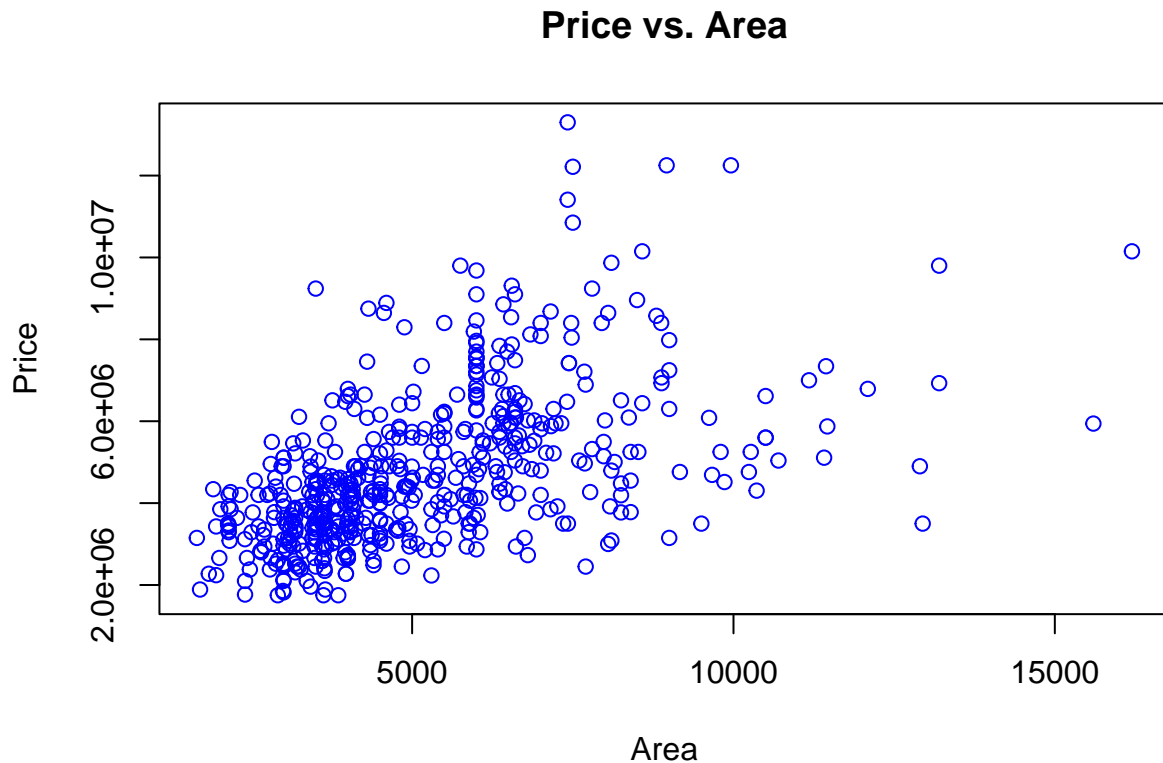
We can see a visible linear correlation between the two variables, suggesting that area will be a large factor in the models we will test.

All in all, this dataset is very interesting to us because of how it can teach us the relationships between a house's price and attributes about the house and can help us better understand the housing market when we need to go out and purchase a house for ourselves.

Section 2: Regression Analysis

Collinearity

In our study of the dataset we must ask ourselves whether or not the data is plagued by the issue of collinearity. Do the predictors present a linear relationship, not just with the response, but within each other as well? Collinearity can hurt a model's performance; inflated R^2 scores and diminished p-values are but a few of the problems that arise alongside predictor collinearity... and so, the dataset must be tested for it.

Let us begin by creating a ***correlation heat-map*** for each predictor. This measures the correlation between predictors, and provides us with a visual method of estimating it. Each categorical column is *label encoded* - their class value is turned to a constant integer - and is added to the correlation matrix (code 4).

```
## Warning: package 'corrplot' was built under R version 4.3.3
```

```
## corrplot 0.92 loaded
```

Correlation heat-map (diagonal labels are the predictor names):

	mainroad	area	parking	prefarea	guestroom	basement	airconditioning	bathrooms	bedrooms	stories	hotwaterheating	furnishingstatus
mainroad	mainroad	0.29	0.2	0.2	0.09	0.04	0.11	0.04	−0.01	0.12	−0.01	−0.16
area	0.29	area	0.35	0.23	0.14	0.05	0.22	0.19	0.15	0.08	−0.01	−0.17
parking	0.2	0.35	parking	0.09	0.04	0.05	0.16	0.18	0.14	0.05	0.07	−0.18
prefarea	0.2	0.23	0.09	prefarea	0.16	0.23	0.12	0.06	0.08	0.04	−0.06	−0.11
guestroom	0.09	0.14	0.04	0.16	guestroom	0.37	0.14	0.13	0.08	0.04	−0.01	−0.12
basement	0.04	0.05	0.05	0.23	0.37	basement	0.05	0.1	0.1	−0.17	0	−0.11
airconditioning	0.11	0.22	0.16	0.12	0.14	0.05	airconditioning	0.19	0.16	0.29	−0.13	−0.15
bathrooms	0.04	0.19	0.18	0.06	0.13	0.1	0.19	bathrooms	0.37	0.33	0.07	−0.14
bedrooms	−0.01	0.15	0.14	0.08	0.08	0.1	0.16	0.37	bedrooms	0.41	0.05	−0.12
stories	0.12	0.08	0.05	0.04	0.04	−0.17	0.29	0.33	0.41	stories	0.02	−0.1
hotwaterheating	−0.01	−0.01	0.07	−0.06	−0.01	0	−0.13	0.07	0.05	0.02	hotwaterheating	−0.03
furnishingstatus	−0.16	−0.17	−0.18	−0.11	−0.12	−0.11	−0.15	−0.14	−0.12	−0.1	−0.03	furnishingstatus

As we can see by the correlation heat-map, correlations between each predictor are realtively small. The strongest correlation is between *stories* and *bedrooms*, with a value of 0.41, swiftly followed by *bedrooms* and *bathrooms* with a coefficient of 0.37, and by *parking* and *area*, with a value of 0.35. These coefficients are decently small, however we will proceed with a more thorough investigation to make sure collinearity is not one of this dataset's problems.

Our next step is viewing the dataset's condition numbers, which are listed below (code 5):

Eigenvalue	Condition Index
11.4591	1.0000
0.5846	4.428
0.2140	7.318
0.1537	8.633
0.1229	9.655
0.1016	10.622
0.0886	11.372
0.0778	12.136
0.0622	13.574
0.0569	14.194
0.0403	16.867
0.0309	19.272
0.0075	38.997

The condition number is 38.997, and usually a value above 30 marks the dataset for collinearity. This value suggests that collinearity should be a problem in this dataset. To further confirm this hypothesis, we shall check each predictor's VIF value (code 6).

4

Variable	VIF
area	1.325
bedrooms	1.368
bathrooms	1.287
stories	1.478
mainroad	1.173
guestroom	1.213
basement	1.321
hotwaterheating	1.039
airconditioning	1.207
parking	1.212
prefarea	1.149
furnishingstatus	1.096

Only predictors with VIFs above 5 are considered problematic, however we can see that, in this dataset, the highest VIF value is 1.478. We could stop, however we wish to make the investigation very thorough.

We can then check if the variables are orthogonal. An orthogonal variable is not plagued by multicollinearity, and has a R^2_k of less than 0.3. Below is a table of these values for each predictor (code 7):

Variable	R^2_k
area	0.245
bedrooms	0.269
bathrooms	0.227
stories	0.323
mainroad	0.147
guestroom	0.175
basement	0.243
hotwaterheating	0.038
airconditioning	0.172
parking	0.175
prefarea	0.129
furnishingstatus	0.087

We can see that most predictors are orthogonal, with the exception of *stories* which has a value of 0.323.

If, perchance, there was an issue of collinearity, we could check the individual condition numbers. They would have to be above 30. Each value above 30 would signify a singular instance of a problematic linear dependence between predictors. Luckily, there is none here, as you can see from the values below (code 8):

Eigenvalue	Condition Index	intercept	area	bedrooms	bathrooms	stories
11.459	1.000	0.000	0.001	0.000	0.001	0.001
0.585	4.428	0.000	0.003	0.000	0.000	0.001
0.214	7.318	0.000	0.001	0.004	0.036	0.340
0.154	8.633	0.001	0.070	0.000	0.008	0.008
0.123	9.655	0.000	0.467	0.005	0.108	0.001
0.102	10.622	0.000	0.177	0.004	0.472	0.093
0.089	11.372	0.000	0.011	0.003	0.006	0.083
0.078	12.136	0.000	0.096	0.000	0.180	0.123
0.062	13.574	0.006	0.001	0.046	0.087	0.017
0.057	14.194	0.004	0.121	0.014	0.030	0.138

Eigenvalue	Condition Index	intercept	area	bedrooms	bathrooms	stories
0.040	16.867	0.000	0.000	0.751	0.068	0.147
0.031	19.271	0.003	0.053	0.067	0.002	0.041
0.008	38.997	0.986	0.001	0.106	0.001	0.008

mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0.000	0.001	0.001	0.000	0.001	0.002	0.001	0.001
0.000	0.001	0.001	0.000	0.000	0.775	0.001	0.007
0.001	0.012	0.058	0.001	0.013	0.005	0.014	0.034
0.000	0.030	0.046	0.007	0.011	0.093	0.032	0.423
0.004	0.033	0.128	0.001	0.018	0.052	0.029	0.014
0.001	0.020	0.007	0.003	0.156	0.040	0.058	0.004
0.002	0.029	0.000	0.003	0.475	0.005	0.419	0.013
0.002	0.271	0.001	0.009	0.233	0.000	0.265	0.006
0.027	0.363	0.047	0.157	0.007	0.014	0.097	0.205
0.028	0.184	0.586	0.108	0.005	0.000	0.021	0.105
0.091	0.040	0.107	0.011	0.002	0.000	0.012	0.000
0.519	0.001	0.010	0.404	0.020	0.001	0.041	0.003
0.325	0.015	0.009	0.296	0.060	0.014	0.010	0.184

With all the information provided, we then conclude, with confidence, that multicollinearity is not a problem of this dataset.

Variable Selection

We now move on to selecting variables to use in our final model. We will perform this using forward, backward, and step-wise selection (code 9), using both AIC and BIC, and reporting the LOO-CV RMSE and adjusted R^2 for each model to determine which model is the best (code 10).

Model	LOO-CV RMSE	Adjusted R^2
Backwards AIC	1087331	0.6740117
Backwards BIC	1090310	0.6706188
Forwards AIC	1087331	0.6740117
Forwards BIC	1090310	0.6706188
Step-wise AIC	1087331	0.6740117
Step-wise BIC	1090310	0.6706188

After finding all of the values for LOO-CV RMSE and adjusted R^2 for each model, we see that the best models are the backwards, forwards, and step-wise models that used AIC as their metric. All three of these models are the full model with no variables removed. Therefore, we will stick with the model containing all 12 of the original predictors and move on to model diagnostics.

Model Diagnostics

In order to remove problematic data points from the data set, we must perform a series of tests to identify them. To begin, we will examine our error assumptions to see if any have been violated using graphical methods and hypothesis tests.

Constant Variance Assumption To verify that our model follows the constant variance assumption, we will be using the Breush-Pagan test (code 11) at a significance level of $\alpha = .05$. Our null and alternative hypotheses are H_0: Homoscedastic errors and H_1: Heteroscedastic errors.

```
##
##  studentized Breusch-Pagan test
##
## data:  model
## BP = 68.416, df = 13, p-value = 1.569e-09
```
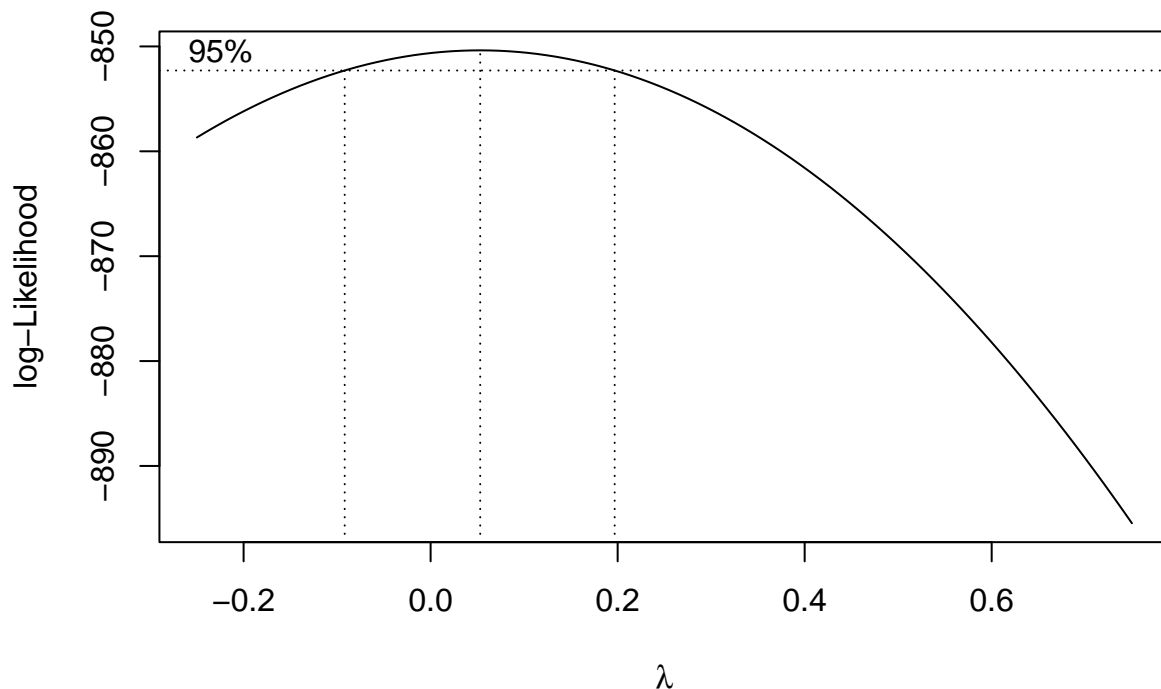
The value of our test-statistic is 68.416 and our p-value is $1.569 \times e^{-9}$. We reject the null hypothesis at the $\alpha = .05$ significance level and conclude that constant variance assumption is violated.

Normality Assumption In order to check the normality assumption, we will be performing the Shapiro-Wilk test (code 12) at the $\alpha = .05$ significance level. Our null and alternative hypotheses are H_0: The errors are normally distributed and H_1: The errors are not normally distributed.

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(model)
## W = 0.95399, p-value = 5.31e-12
```

The value of our test-statistic is $.95399$ and our p-value is $5.31 \times e^{-12}$. We reject the null hypothesis at the $\alpha = .05$ significance level and conclude that errors are not normally distributed.

In order to try to correct for this, we will be using the Box-Cox method (code 13).

```
## [1] 0.0530303
```

According to the plot, and the print out, we can tell that $\hat{\lambda} = .05303$. So we will be performing an OLS regression with price$^{.05303}$ as the response and performing the Shapiro-Wilk test at the $\alpha = .05$ significance level (code 14).

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(model_bc)
## W = 0.99525, p-value = 0.0939
```

The value of our test-statistic is 0.9953 and our p-value is 0.0939. We accept the null hypothesis at the $\alpha = .05$ significance level and conclude that the errors are now normally distributed.

We will also check the RMSE and the percent variation in price explained by both models (code 15).

Model	RMSE	% Variation
Standard Model	1054129	68
Transformed Model	1041366	69

Since our errors are now normally distributed, we will use the transformed model instead of the standard model.

Highly Influential Points   To check for highly influential points, we will be checking the cooks distances of the data (code 16).

```
## [1] 35
```

Using this distance, we find that there are 35 highly influential points. We will check to see if removing these points from the model will in any way correct our models assumption violations. First we will check the constant variance assumption at the $\alpha = .05$ significance level (code 17).

```
##
##  studentized Breusch-Pagan test
##
## data:  model_fix
## BP = 31.917, df = 13, p-value = 0.002471
```

The value of our test-statistic is 31.917 and our p-value is .00247. We reject the null hypothesis at the $\alpha = .05$ significance level and conclude that constant variance assumption is violated.

Next we will check our constant variance assumption at the $\alpha = .05$ significance level (code 18).

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(model_fix)
## W = 0.99865, p-value = 0.9681
```

The value of our test-statistic is .9987 and our p-value is .9681. We accept the null hypothesis at the $\alpha = .05$ significance level and conclude that errors are still normally distributed.

We will also check the R^2 of both models (code 19),

```
## [1] 0.6818018
```

```
## [1] 0.7651252
```

Model	R^2
Standard Model	.682
Model with Influential Points Removed	.765

It is clear that the Model with Influential Points removed is the preferred model as it explains almost 8% more of the observed variability in price with the predictors.

Outliers To check for outliers we will be using the studentized residuals at the $\alpha = .05$ significance level (code 20).

```
## [1] 0
```

We can see that there are zero outliers, so we move on. Our model now has normally distributed errors, however the equal variance assumption is violated. Therefore, we will look to use a regression method other than ordinary least squares. Since our errors are not correlated (heteroscedastic) we will use weighted least squares.

Weighted Least Squares (WLS)

We now begin performing the weighted least squares regression. We create a model using the absolute values of the residual and then calculate weights as $\frac{1}{fittedvalues^2}$. We then create our model with these weights (code 21).

After creating our model, we will perform a t-test at the 5% significance for each parameter for the OLS model and the WLS model (code 22).

Below are the estimates and p-values for the OLS regression:

Coefficient	Estimate	P-value
Intercept	2.14	0.000
area	5.91e-06	2.69e-28
bedrooms	0.00307	0.0361
bathrooms	0.0203	1.74e-19
stories	0.0111	6.06e-17
mainroadyes	0.0111	0.000110
guestroomyes	0.00833	0.00206
basementyes	0.00896	0.000086
hotwaterheatingyes	0.0182	0.000538
airconditioningyes	0.0215	5.06e-21
parking	0.00519	0.0000168
prefareayes	0.0158	4.39e-11

Coefficient	Estimate	P-value
furnishingstatussemi-furnished	0.000412	0.862
furnishingstatusunfurnished	-0.0135	1.53e-07

Below are the estimates and p-values for the WLS regression:

Coefficient	Estimate	P-value
Intercept	2.140940e+00	0.000000e+00
area	5.801192e-06	1.616203e-30
bedrooms	2.901672e-03	3.903794e-02
bathrooms	2.021508e-02	6.941308e-25
stories	1.031211e-02	1.608922e-16
mainroadyes	1.292915e-02	3.337035e-07
guestroomyes	8.219860e-03	8.126159e-04
basementyes	7.518584e-03	2.664939e-04
hotwaterheatingyes	1.890570e-02	2.471119e-06
airconditioningyes	2.060580e-02	3.670600e-23
parking	4.690907e-03	4.059220e-05
prefareayes	1.569487e-02	1.598082e-13
furnishingstatussemi-furnished	1.335054e-03	5.229621e-01
furnishingstatusunfurnished	-1.313332e-02	2.004748e-07

For both OLS and WLS, every variable is significant except furnishing status semi-furnished.

We now summarize both the OLS model and WLS model to see the R^2 scores and how they compare (code 23).

Model	R^2	RMSE
OLS	0.765	824504.7
WLS	0.786	843030.6

After finding the R^2 score and RMSE of each regression, we see that WLS has a better R^2 score and OLS has a better RMSE. We recall that our goal of this report is to find the model which best explains the variance in the price of houses. Therefore, we will choose the WLS model as the better model because it has a higher R^2 score and thus a higher percentage of the variance of price being explained by the model (code 24).

After performing collinearity tests, variable selection, model transformations, model diagnostics, and WLS, we are left with our final regression model with the final equation:

$$\text{Price}_i^{.05303} = 2.14 + 0.00000580\,\text{area}_i + 0.00290\,\text{bedrooms}_i + 0.0202\,\text{bathrooms}_i + 0.0103\,\text{stories}_i + 0.0129\,\text{mainroadyes}_i$$

$$+\,0.00822\,\text{guestroomyes}_i + 0.00752\,\text{basementyes}_i + 0.0189\,\text{hotwaterheatingyes}_i + 0.0206\,\text{airconditioningyes}_i$$

$$+0.00469\,\text{parking}_i + 0.0157\,\text{prefareayes}_i + 0.00134\,\text{furnishingstatus-semi-furnished}_i - 0.0131\,\text{furnishingstatus-unfurnished}_i$$

Section 3: Discussion

Our final model has all 12 of the original predictors, has transformed the response as being $\text{Price}_i^{.05303}$, has removed points that are deemed highly influential, and has applied the weights of $\frac{1}{fittedvalues^2}$ to perform a

weighted least squares regression. Our original, standard OLS model had an R^2 score of 0.68 and our final model has an R^2 score of 0.786. We see that this model is much better than our original model and is far better at explaining the variance in price of each house. We also notice that our prediction that the variable "area" would be a large factor in the model is correct, as we can see it has a far lower p-value than any other variable in the model (1.616203e-30). With this revised model, we can more accurately explain the change of the prices of houses from this dataset when we are able to observe attributes of the house that we worked with in this dataset. This model can be used to predict the prices of houses in India if we had access to the 12 predictors that were used in this dataset and is a good way to learn the trends of the Indian housing market as it correlates to features of a house.

Section 4: Limitations

There were some issues with the data set. The first is that we were expecting collinearity in the variables stories and area, however there was no collinearity between these variables. This points to the fact that the data may be flawed in some way. Perhaps the dataset is artificially created or random since in real life the number of stories and total area of a house should be correlated in some way. Also, we would add more columns to this data set such as location, proximity to major city, and the average income of each location. These variables are important because a house near the city will cost much more than a house in a rural area. It would be easier to predict housing prices given the location of the house and the proximity to a major city. Another limitation of our dataset came from the predictor "prefarea". We don't know what defines an area to be "preferred", so there is no telling what this variable means. Also, as noted earlier, since we can get a good grasp on the precise location these houses are in, we can't advise that our model can be used for real-life predictions because every region has different housing markets.

Section 5: Conclusions

This study successfully identified key predictors that influence house prices using a comprehensive dataset of 545 observations in India. Through rigorous statistical testing and modeling, we concluded that variables such as area, number of bathrooms, air conditioning, and location in a preferred area significantly impact house prices. The robustness of our results was ensured by addressing potential multicollinearity, which was not found to be problematic in our dataset.

From the model diagnostics, it became evident that the assumptions of homoscedasticity and normality were initially violated. However, through transformations and the removal of influential points, these issues were substantially mitigated, leading to a more reliable model. Ultimately, the Weighted Least Squares (WLS) model proved superior, explaining a higher percentage of variance in house prices than the Ordinary Least Squares (OLS) model.

Future Work: Expanding the Dataset: Further research could benefit from including more variables, such as the age of the house, recent renovations, and proximity to essential services, which might provide deeper insights into price determinants.

- Advanced Modeling Techniques: Employing more sophisticated regression techniques or machine learning models could potentially uncover non-linear relationships or interactions between predictors that were not explored in this study.

- Cross-Validation: Implementing more rigorous cross-validation techniques could enhance the generalizability of the model to other regions or different housing markets.

This project not only enhanced our understanding of the housing market dynamics but also equipped us with valuable modeling techniques that can be applied to other economic sectors.

Section 6: Additional Work

Some additional work to add is the performing of variable selection again after transforming our model and removing influential points. We will run step-wise variable selection using BIC as our metric and report the adjusted R^2 of new model versus our previous model (code 25).

Model	Adjusted R^2
Original Model	0.759
New Model (after variable selection #2)	0.757

As we can see, our metric has not improved, hence why we didn't include a second round of variable selection.

Code Appendix

Code 1: Printing the first 5 rows of our dataset.

```
data <- read.csv('Housing.csv')
head(data, 5)
```

Code 2: Creating a histogram for price as well as printing statistical information.

```
library(dplyr)
hist(data$price, xlab="Price of House", ylab="Frequency",
     main='Histogram of Price' ,col="purple")
price_summary <- data %>%
  summarise(
    Min = min(price, na.rm = TRUE),
    Q1 = quantile(price, 0.25, na.rm = TRUE),
    Median = median(price, na.rm = TRUE),
    Mean = mean(price, na.rm = TRUE),
    Q3 = quantile(price, 0.75, na.rm = TRUE),
    Max = max(price, na.rm = TRUE),
    NA_Count = sum(is.na(price))
  )
print(price_summary)
```

Code 3: Plotting a price vs. area scatter plot.

```
plot(data$area, data$price, xlab='Area', ylab='Price', main='Price vs. Area', col='blue')
```

Code 4: Creating a correlation heatmap.

```
library(dplyr)
library(corrplot)
library(forcats)
data <- data %>%
  mutate(across(where(is.character), as.factor))
data_encoded <- data %>%
  mutate(across(where(~is.factor(.) && nlevels(.) <= 3), as.integer))
data_numeric_only <- data_encoded %>%
```

```
   select(where(is.numeric))
housing_preds <- select(data_numeric_only, -price)
cor_matrix <- cor(housing_preds, use = "complete.obs")
corrplot(cor_matrix, method = 'color', order = 'hclust', diag = FALSE,
         number.cex = 0.7, addCoef.col = 'black', tl.pos = 'd', tl.cex = 0.6,
         cl.cex = 0.7, cl.pos = 'r')
```

Code 5: Printing the eigenvalues and condition numbers of the model.

```
pairs(housing_preds, col = 'dodgerblue', pch=1)
library(olsrr)
model = lm(price ~ ., data = data_numeric_only)
round(ols_eigen_cindex(model)[, 1:2], 4)
```

Code 6: Printing the VIF values for each predictor.

```
library(faraway)
vif(model)
```

Code 7: Printing the R_k^2 values for each predictor.

```
1 - 1/vif(model)
```

Code 8: Printing the eigenvalues and condition numbers for each predictor.

```
round(ols_eigen_cindex(model), 3)
```

Code 9: Performing forward, backward, and step-wise model selection.

```
n = nrow(data)
mod_all_preds = lm(price~., data=data)
mod_back_aic = step(mod_all_preds, direction='backward')
n = nrow(data)
mod_all_preds = lm(price~., data=data)
mod_back_bic = step(mod_all_preds, direction='backward', k=log(n))
n = nrow(data)
mod_start = lm(price~1, data=data)
mod_forwd_aic = step(mod_start, scope=price~mainroad+area+parking+prefarea
                     +guestroom+basement+airconditioning+bathrooms+bedrooms+stories
                     +hotwaterheating+furnishingstatus, direction='forward')
n = nrow(data)
mod_start = lm(price~1, data=data)
mod_forwd_bic = step(mod_start, scope=price~mainroad+area+parking+prefarea
                     +guestroom+basement+airconditioning+bathrooms+bedrooms+stories
                     +hotwaterheating+furnishingstatus, direction='forward', k=log(n))
n = nrow(data)
mod_start = lm(price~1, data=data)
mod_step_aic = step(mod_start, scope=price~mainroad+area+parking+prefarea+guestroom
                     +basement+airconditioning+bathrooms+bedrooms+stories
                     +hotwaterheating+furnishingstatus, direction='both')
n = nrow(data)
```

```
mod_start = lm(price~1, data=data)
mod_step_bic = step(mod_start, scope=price~mainroad+area+parking+prefarea+guestroom
                    +basement+airconditioning+bathrooms+bedrooms+stories
                    +hotwaterheating+furnishingstatus, direction='both', k=log(n))
```

Code 10: Printing the LOO-CV RMSE and Adjusted R^2 for each model.

```
calc_loocv_rmse = function(model) {
  sqrt(mean((resid(model) / (1 - hatvalues(model))) ^ 2))
}
calc_loocv_rmse(mod_back_aic)
calc_loocv_rmse(mod_back_bic)
calc_loocv_rmse(mod_forwd_aic)
calc_loocv_rmse(mod_forwd_bic)
calc_loocv_rmse(mod_step_aic)
calc_loocv_rmse(mod_step_bic)
summary(mod_back_aic)$adj.r.squared
summary(mod_back_bic)$adj.r.squared
summary(mod_forwd_aic)$adj.r.squared
summary(mod_forwd_bic)$adj.r.squared
summary(mod_step_aic)$adj.r.squared
summary(mod_step_bic)$adj.r.squared
```

Code 11: Performing a Breusch-Pagan test.

```
library(olsrr)
library(lmtest)
model = lm(price = ., data = data)
bptest(model)
```

Code 12: Performing a Shapiro-Wilk test.

```
shapiro.test(resid(model))
```

Code 13: Creating a Box Cox model and printing the maximum.

```
library(MASS)
bc = boxcox(model, lambda = seq(-0.25, 0.75, by = 0.05), plotit = TRUE)
bc$x[which.max(bc$y)]
```

Code 14: Performing a Shapiro-Wilk test.

```
model_bc = lm(price ^ 0.05303 ~ . , data = data)
shapiro.test(resid(model_bc))
```

Code 15: Printing the R^2 values and RMSE of each model.

```
sqrt(mean((data$price - predict(model))^2))
sqrt(mean((data$price - predict(model_bc)^(1/0.05303))^2))
SST = sum((data$price - mean(data$price))^2)
1 - sum((data$price - predict(model))^2) / SST
1 - sum((data$price - predict(model_bc)^(1/0.05303))^2) / SST
```

Code 16: Printing how many observations are influential.

```r
length(which(cooks.distance(model_bc) > 4 / length(cooks.distance(model_bc))))
```

Code 17: Performing a Breusch-Pagan test on the model without influential points.

```r
noninfluential_ids = which(
    cooks.distance(model_bc) <= 4/ length(cooks.distance(model_bc)))
model_fix = lm(price^ 0.05303 ~ .,
               data = data,
               subset = noninfluential_ids)
bptest(model_fix)
```

Code 18: Performing a Shapiro-Wilk test on the model without influential points.

```r
shapiro.test(resid(model_fix))
```

Code 19: Printing the R^2 value of each model.

```r
summary(model)$r.squared
```

```r
summary(model_fix)$r.squared
```

Code 20: Printing out the number of observations deemed to be outliers.

```r
outlier_test_cutoff = function(model_bc, alpha = 0.05) {
    n = length(resid(model_bc))
    qt(alpha/(2 * n), df = df.residual(model_bc) - 1, lower.tail = FALSE)
}

cutoff = outlier_test_cutoff(model_bc, alpha = 0.05)

length(which(abs(rstudent(model_bc)) > cutoff))
```

Code 21: Performing WLS regression.

```r
model_wts = lm(abs(resid(model_fix)) ~ ., data = data[noninfluential_ids,])
coef(model_wts)
weights = 1 / (fitted(model_wts)^2)
model_wls = lm(price^ 0.05303 ~ ., data=data[noninfluential_ids,], weights = weights)
```

Code 22: Printing the coefficients of the OLS and WLS regressions.

```r
summary(model_fix)$coef
summary(model_wls)$coef
```

Code 23: Printing the R^2 and RMSE of the OLS and WLS regressions.

```r
summary(model_fix)$r.squared
summary(model_wls)$r.squared
sqrt(mean((data[noninfluential_ids,]$price - predict(model_fix)^(1/0.05303))^2))
sqrt(mean((data[noninfluential_ids,]$price - predict(model_wls)^(1/0.05303))^2))
```

Code 24: Printing the final coefficients.

```
summary(model_wls)$coef
```

Code 25: Performing variable selection again on our and refined model.

```
n=nrow(data[noninfluential_ids,])
mod_start2 = lm(price^0.05303~1, data=data[noninfluential_ids,])
mod_step_bic2 = step(mod_start2, scope=price^0.05303~mainroad+area+parking+prefarea
                      +guestroom+basement+airconditioning+bathrooms+bedrooms+stories
                      +hotwaterheating+furnishingstatus, direction='both', k=log(n))
summary(mod_step_bic2)$adj.r.squared
summary(model_fix)$adj.r.squared
```