House Prices

CUNY SPS DATA 621

GROUP 2: William Aiken, Donald Butler, Michael Ippolito, Bharani Nittala, and Leticia Salazar

Abstract

Many dream about their first home, where to buy, how many bedrooms or baths, if it should include a pool or a large backyard, and how much buying a home will cost. With the housing market affected by factors such as mortgage rates, inflation, or economic recession, housing prices have risen dramatically over the years. In this project, we built a model to predict housing prices based on such factors as house area, number of bedrooms, whether it is furnished, nearness to a main road, and the presence of amenities like basements and guest houses. We used a multivariate linear model against a data set of 545 houses. We augmented the model by performing log transformation on the response variable (price) and by Huber-weighting the model by performing robust regression. Based on our model analysis, the basic assumptions of linear modeling are valid, with the exception of the presence of non-normally distributed residuals. However, this is less of an issue if the data set is large enough compared to the number of predictors. The resulting model performed reasonably well, yielding an adjusted R-squared value of 0.710 against our training data and 0.709 against the validation set. Confounding factors may be present but missing from the data set, as there were some outliers that could not be explained by any of the available predictors. Future work could be performed to generate a richer data set with additional explanatory variables.

Key words

house prices, linear models, regression, home investment, real estate

Introduction:

The housing market pricing changes can be affected by factors such as mortgage rates, inflation, or economic recessions, to name a few. From 1950 to the present, we witness home price fluctuations, with some steady rising periods between 1950s to the late 1980s. According to the article from Better.com (Johnson, 2022), if we start analyzing the median home value we note that house price growth has risen 326.1% since 1950 with a current median home price of $336,900 (inflation-adjusted to 2020 dollars). There was a 90% drop in production of new homes before the Great Depression started, and, in 1956, President Dwight D. Eisenhower passed the Federal Aid Highway Act that enabled the possibility to access a suburban way of life. By 1960, the boom of home ownership caused home prices started to increase. The Federal Housing Administration (FHA) created changes that allowed Americans to afford homes after WWII which contributed to the rise of home ownership and prices. With a median home price increase of 42.8% by the 1970s, the US economy saw inflation rising and, with it, the housing market. Fast forward to the 1980s, when there were high unemployment rates that caused mortgage interest rates to rise even more. During this time, the median home price (inflation-adjusted to 2020 dollars) had an average interest rate for a 30-year fixed mortgage of 13.7%. By the 1990s, the recession had ended and the housing bubble burst affected only major cities like New York, Boston, Los Angeles, San Diego, Washington D.C., and San Francisco. Entering the 21st century, the average interest rate for a 30-year fixed mortgage was down to 8.1%, as another recession was not expected. By 2008, the US faced challenges that pushed more than 10 million Americans to lose their homes and 9 million to lose their jobs. In 2020, the average interest rate for a 30-year fixed mortgage was

3.1%, with a record-setting year for housing prices. Now we can't forget the pandemic of COVID-19 that left many to work from home. How does this contribute to the housing prices? Due to the pandemic, housing prices rose once again only to drop in some cities where the demand for homes wasn't as high. It is believed that housing prices will soon level off and slow down but that mortgage rate will continue to rise.

Literature Review

While our assessment didn't focus on real estate prices on a longitudinal scale, we evaluated the factors that contribute to prices at a single instant in time. Using various predictors, we built a multivariate linear model to predict housing prices based on a dataset we obtained from Kaggle.com (H., 2022). This approach is not new. Multivariate linear modeling to predict housing prices was done previously by Manjula (2017), Man (2017), and others. Our review of current predictive methodologies indicates a number of other approaches have been taken, including machine-learning (Cuturi & Etchebarne, 2021) and sequential models (Majumder, 2022). We limited our approach to constructing a multivariate linear model from a single dataset using a lean set of predictors.

Methodology

Our dataset consists of 545 records obtained from Kaggle (H, 2017). First, we assessed the data to get an insight on what we were working with and to later determine what multiple linear regression model would be appropriate to use. We examined various predictors to evaluate qualitatively how they might affect our response variable (price). We also compared predictors against each other to look for signs of colinearity. Our data consists of both quantitative and categorical variables, the latter of which required factoring to use in our modeling. To prepare for modeling, we split our data into a training and evaluation set in order to properly make predictions. We then took the approach of attempting a linear model against the full dataset, followed by stepwise reducing the predictors to a leaner set. At each stage, we examined the adjusted R-squared value to determine model fit. We also evaluated the model to validate the underlying assumptions upon which linear models are based, namely:

- Normality of residuals
- Homoschedasticity of residuals
- Independence of predictors
- Linearity of fit

To evaluate normality of residuals, we used the Shapiro test for normality. We used the Breusch-Pagan test to evalute homoschedasticity of residuals. In addition, we examined colinearity problems by looking at variance inflation factors (VIFs). We also evaluated outliers by looking at hat values and comparing them to the a cutoff point based on 2 * (p + 1) / n, where p is the number of parameters and n is the number of observations

Based on our initial modeling attempts, there were no problems with linearity of fit or colinearity among the predictors. However, it was evident that the residuals were heteroschedastic and not normally distributed. We attempted to correct for these problems by applying a log transformation to the response variable (price). While this did yield homoschedastic residuals, they remained non-normally distributed. To attempt to correct for this, we attempted a Huber-weighted robust model. While this didn't fully correct the issue, we note that in larger datasets, non-normality of residuals doesn't significant affect model results (Schmidt, 2018), particularly those in which the number of records exceeds ten times the number of predictors. We also examined outliers to evaluate whether there was any trend evident in the data. However, we could discern none, and, therefore, we removed the most significant outliers to attempt a better fit.

Experimentation and Results

Once we started our data exploration process we found the following:

- The data is composed of 545 observations and 13 variables (Table 1)
- There were no missing values
- We had to split the data into 70-30 ratio to be able to have a training and evaluation data sets.
- This dataset contains homes with prices above 1 million.

We looked into each variable individually and found the following:

- Based on the distribution of the number of bedrooms, it may be best to categorize these with dummy variables; 2, 3, and 4+.
- Based on the distribution of the number of bathrooms, it may be best to categorize these with dummy variables; 2, and 3+.
- Based on the distribution of stories, it would seem to make sense to classify homes with 3 or more floors together by introducing dummy variables; 2, and 3+.
- It would initially seem to make sense to introduce dummy variables; 1, and 2+.
- The furnishing status variable is taking on three values; unfurnished, semi-furnished, and furnished. Since we would consider unfurnished as the default state, we will use dummy variables; semi-furnished and furnished.
- The main road variable is yes/no based on the street of the home. We will replace this with a dummy variable.
- The guest room variable is yes/no based on the home having a guest room. It is unclear from the dataset source if this is in addition to the number of bedrooms, but we would expect houses with a guest room to have a higher price. We will replace this with a dummy variable.
- The basement variable is yes/no based on the home having a basement. It is unclear if having a basement or not would lead to an increase in home price, but we will replace this with a dummy variable for analysis.
- Based on the distribution, we assume that the hot water heating variable represents if the house has in-floor heating, rather than forced air. Based on this assumption, we assume that having this feature would lead to higher house price. The variable will be replaced with a dummy variable for analysis.
- The air conditioning variable indicates if the house has central air conditioning. We would expect homes with air conditioning would have a higher price than those without. The variable will be replaced with a dummy variable.
- The dataset source doesn't specify exactly what the "prefarea" variable represents. We are assuming that this is a yes/no value indicating whether the house is in a preferred neighborhood. We would expect houses with a yes to be higher price than those not.

Once we were ready to start with the data preparation process and knowing there were no missing variables, we introduced a cleaning function to replace our categorical variables with the dummy values. This also ensured that our test and train datasets are processed in the same way. Plots of the cleaned training set were prepared to visually evaluate the data.

Histograms of price and area exhibited a noticeable right skew (Figures 1 and 2, respectively), indicating that the data set contains homes that are smaller and lower in value. A correlation plot (Figure 3) of predictor variables indicates no obvious signs of colinearity. As shown in the figure, price increases with square footage, which makes intuitive sense. Boxplots (Figures 4 and 5) show that pricier homes had guestrooms, basements, air conditioning, four bedrooms, four stories, two-car garages, and two or more bathrooms. It also made sense that unfurnished homes in preferred areas had significantly higher median values. Surprisingly, houses with five and six bedrooms had median values below houses with only four bedrooms. It was also surprising that homes with three-car garages had median prices below those with only two-car garages. Also

counter-intuitively, house on a main road increased the median home value. This could indicate that we misinterpreted the variable and that it perhaps indicates that a main road is only readily accessible, rather than in close proximity. A correlation plot (Figure 6) visually confirms the relationships between variables.

The model-building process started with three models created.

- Model 1 `lm_mod1` was run against the whole data set and yielded an Adjusted R-squared of 0.6755
- Model 2 `lm_mod2` used stepAIC and generated an Adjusted R-squared of 0.6767
- Model 3 `lm_mod3` was created to reduce colinearity and remove low values and yielded an Adjusted R-squared of 0.6703

We developed a function to validate the underlying assumptions of the linear model and evaluated each model against the function. We concluded that residuals were neither normally distributed nor homoschedastic, presenting a problem with the underlying assumptions (Figures 7 and 8). This is confirmed quantitatively using the Shapiro test for normality, which yielded a p-value of $1.33\text{x}10^{11}$. We used the Breusch-Pagan test for homoschedasticity which also yielded a low p-value ($9.25\text{x}10^{8}$), confirming that the residuals are heteroschedastic.

Because of these issues, we chose to run a Box-Cox transformation to give us an idea of whether a transformation of the response variable would be appropriate. The Box-Cox transform resulted in a value of 0.082 on the price variable, indicating that a log transformation would be appropriate. A new model was run and step-reduced, transforming price to use a log scale. While the Shapiro test for normality yielded a higher p-value (0.0274), it still indicated that the residuals were not normally distributed. However, the Breusch-Pagan test resulted in a p-value of 0.221, indicating that the residuals were homoschedastic, a key validation factor. Further, the highest VIF was 2.04, indicating only slight colinearity among some variables. Figure 9 shows the residual analysis visually.

To evaluate the non-normality issue, we examined the most significant outliers (Table 2). There were no obvious indications of a pattern that may have reflected some problem with the model. This could indicate that there are one or more additional variables at play that influence price but are not present in our data set, for example high-end appliances or fixtures, the presence of a pool, or the condition of the property. To attempt to correct for the problem of non-normality, we removed the ten most significant outliers and reran the model, again step-reducing the model to remove non-significant predictors. Unfortunately, the residuals remained non-normally distributed (Figure 10).

In a further attempt to account for the normality problem, we ran a robust linear regression model against the dataset, which yielded a set of weights that we plugged into our linear model. Most weights were 1.0 (Figure 11), indicating that the majority of observations did not need any weighting. The model yielded a significant increase in adjusted R-squared (0.71 compared with 0.68 on the previous run). However, we still had the problem of non-normality of residuals (Figure 12). We tried again to remove the most significant outliers, but that neither increased the adjusted R-squared value nor removed the non-normality issue (Figure 13). To evalutate whether this would invalidate our model, we consulted the literature and found that there is good precedent for ignoring non-normally distributed residuals in datasets that have at least 10 observations per variable (Schmidt, 2018). Since our dataset contains at least that many observations, we can reasonably conclude that the problem of non-normality can be ignored for the purposes of our evaluation.

Based on this assumption, we performed five-fold cross-validation of the model. The adjusted R-squared value remained at 0.71. The final model used a log transformation of price against the following significant predictors:

- area
- mainroad
- guestroom
- basement

- bed3
- bed4
- bath2
- floor2
- floor3
- floor4plus
- car1
- car2
- semifurnished
- furnished
- ac
- neighborhood

Using the final model parameters, we generated the predicted prices in the training data and graphed the data against the actual price (Figure 14). This shows a visually good fit. We then repeated the procedure against the validation dataset, analyzed the residuals (Figure 15), and generated an adjusted R-squared value of 0.71, confirming that our model performed equally well against the validation set. Figure 16 visually confirmed the fit. Table 3 summarizes the results of our model runs.

Discussion and Conclusions As a result of our work, we were able to develop a model that performed reasonably well under training conditions and only slightly poorer for the validation set ((0.710 adjusted R-squared alues of 0.710 and 0.709, respectively). The best linear model used a log transformation on price and 16 predictor variables, weighted using Huber weights determined by robust regression. While the residuals were not normally distributed, the assumptions of linearity, independence of variables, and homoschedasticity of residuals were validated.

We found we were somewhat limited by the data set, in that there appeared to be one or more confounding variables that influenced price and that were not present in the data set. Future research could be done to obtain a richer set of data with additional explanatory variables. That could also mitigate the other limitation we encountered: that of non-normally distributed residuals. However, based on our literature review, that limitation may be less important, as our data set was fairly large compared to the number of explanatory variables.

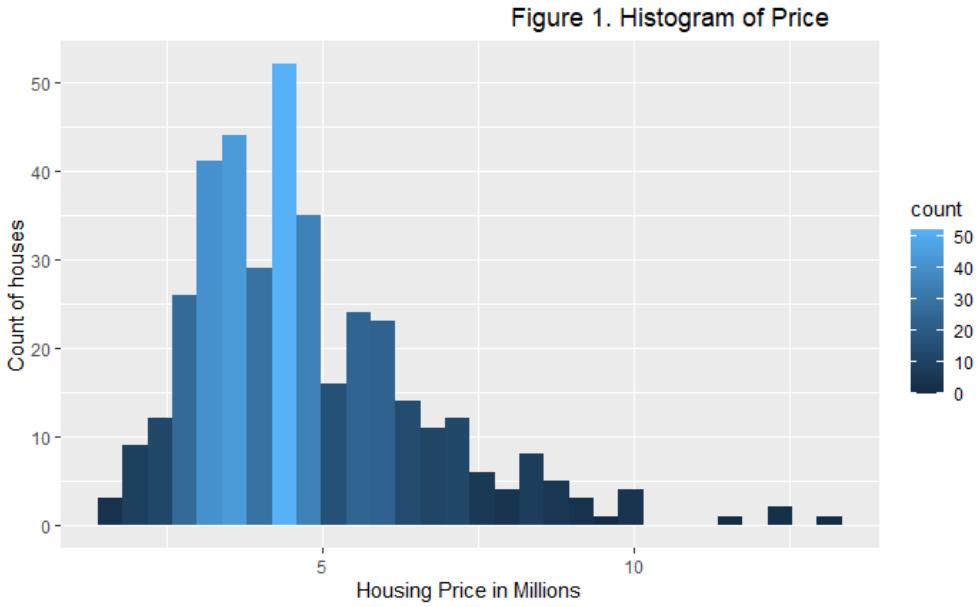
References / Bibliography

- H, M. Y. (2022, January 12). Housing prices dataset. Kaggle. Retrieved November 28, 2022, from https://www.kaggle.com/datasets/yasserh/housing-prices-dataset
- Schmidt, A. (2018, June). Linear Regression and the normality assumption. Retrieved December 5, 2022, from https://doi.org/10.1016/j.jclinepi.2017.12.006
- Johnson, N. (2022, May 24). How much home prices have risen since 1950: Better Mortgage. Better Mortgage Resources. Retrieved December 5, 2022, from https://better.com/content/how-much-home-prices-have-risen-since-1950/
- Majumder, P. (2022, January 27). Using Sequential Model to Predict Prices of Real Estate. Retrieved December 5, 2022, from https://www.analyticsvidhya.com/blog/2022/01/using-sequential-model-to-predict-prices-of-real-estate/
- Cuturi, M. & Etchebarne, G. (2021, June 25). Real Estate pricing with Machine Learning & non-traditional data sources. Retrieved December 5, 2022, from https://tryolabs.com/blog/2021/06/25/real-estate-pricing-with-machine-learning--non-traditional-data-sources

- Man, H. (2017, February 17). Prediction Model-Building a house price model. Retrieved December 5, 2022, from https://medium.com/hanman/data-modeling-building-a-house-price-prediction-model-1450f825073b
- Manjula, R. et al. (2017). Real estate value prediction using multivariate regression models. Retrieved December 5, 2022, from file:///C:/Users/micha/Downloads/Real_estate_value_prediction_using_multivariate_re.pdf

Appendices:

Appendix A. Figures:



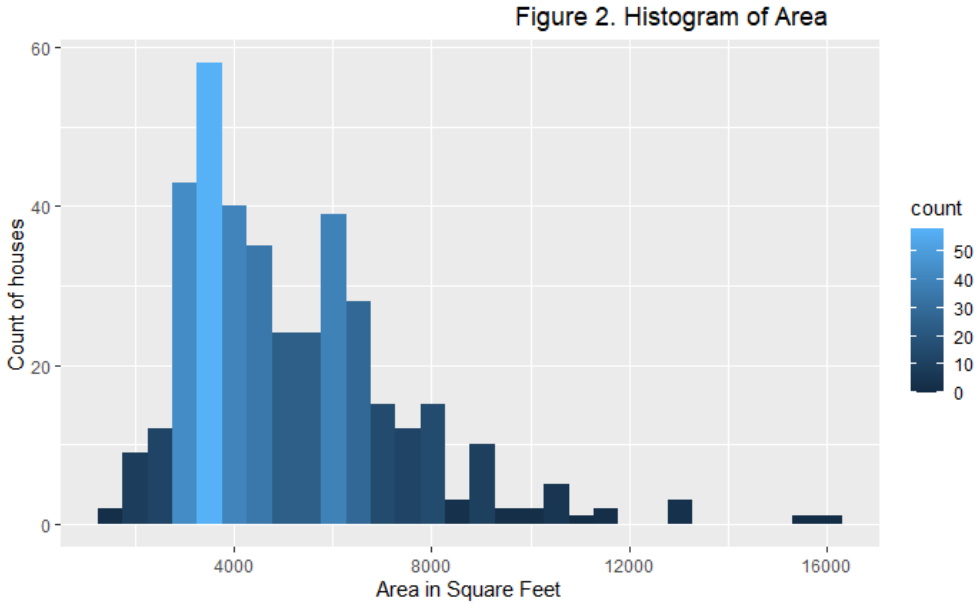


Figure 3. Comparison of quantiative variables

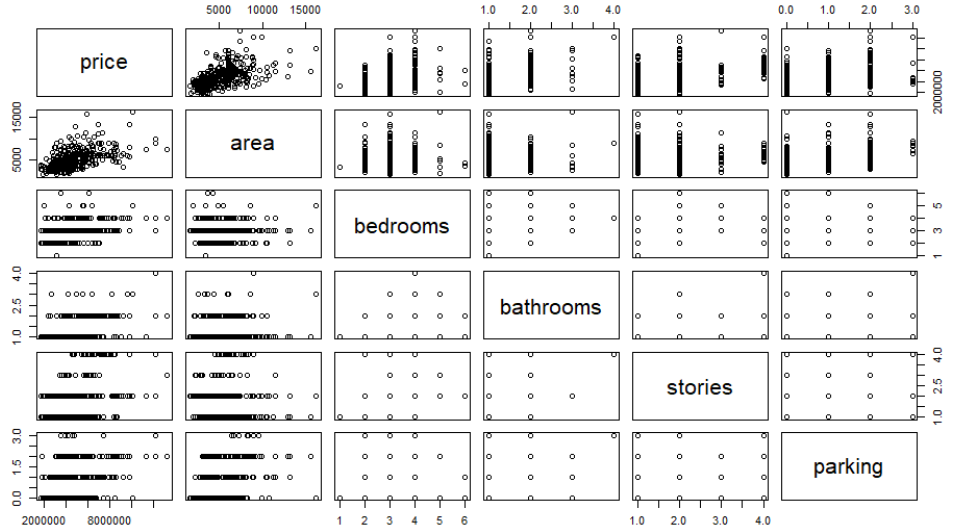


Figure 4. Comparison of nominal categorical variables

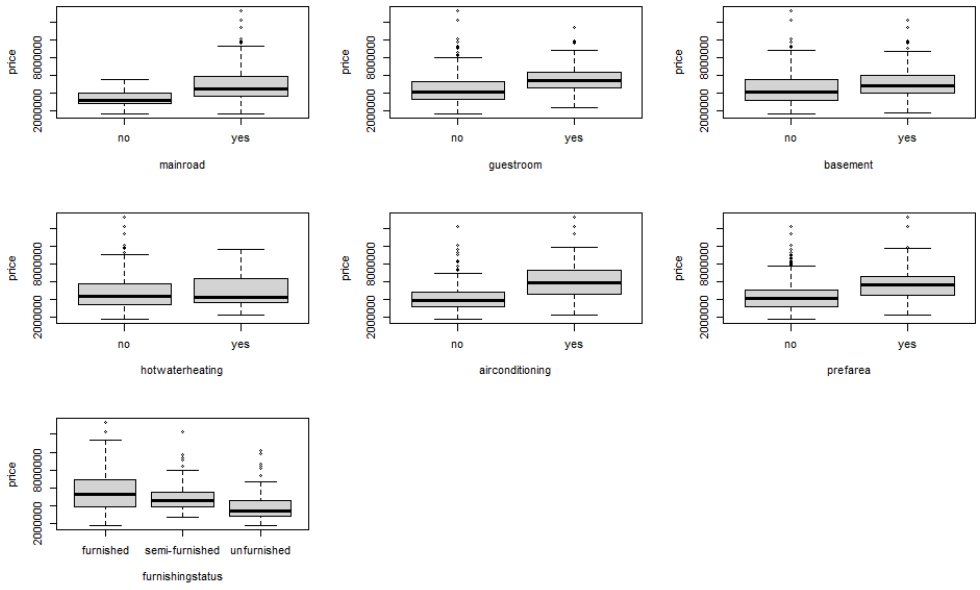


Figure 5. Comparison of ordinal categorical variables

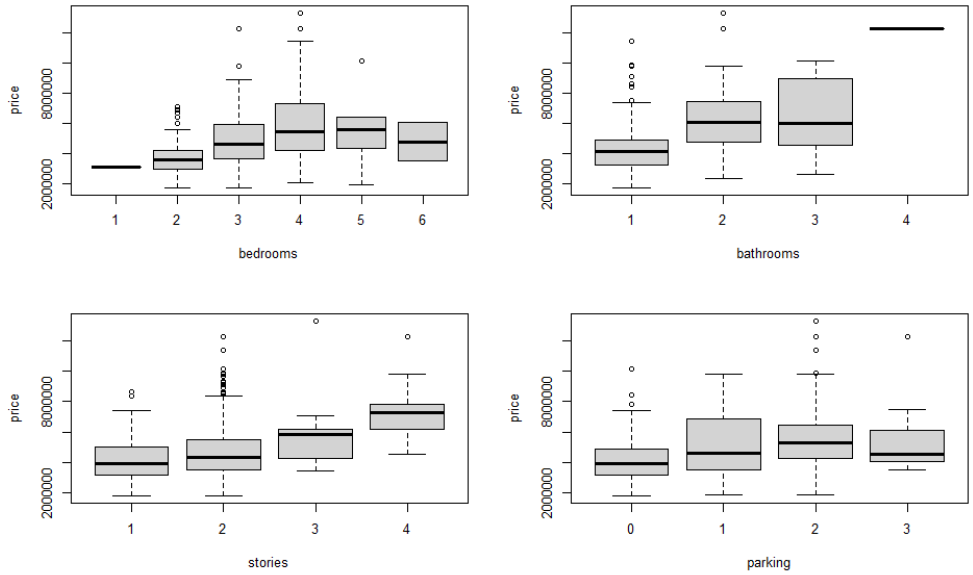


Figure 6. Correlation Plot

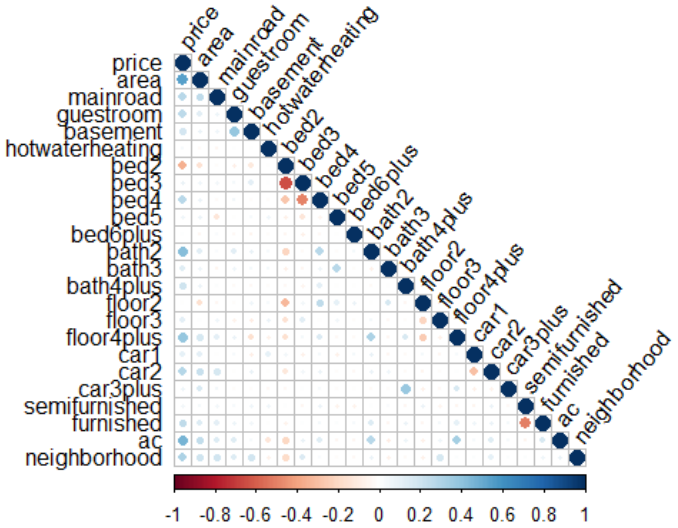


Figure 7. Residual Analysis - Model 2

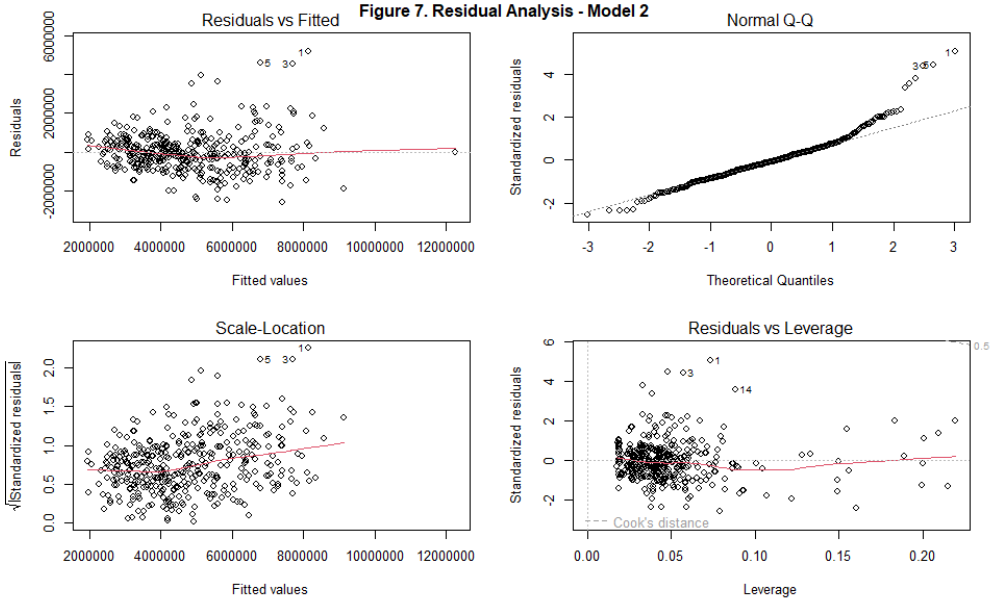


Figure 8. Residual Analysis - Model 3

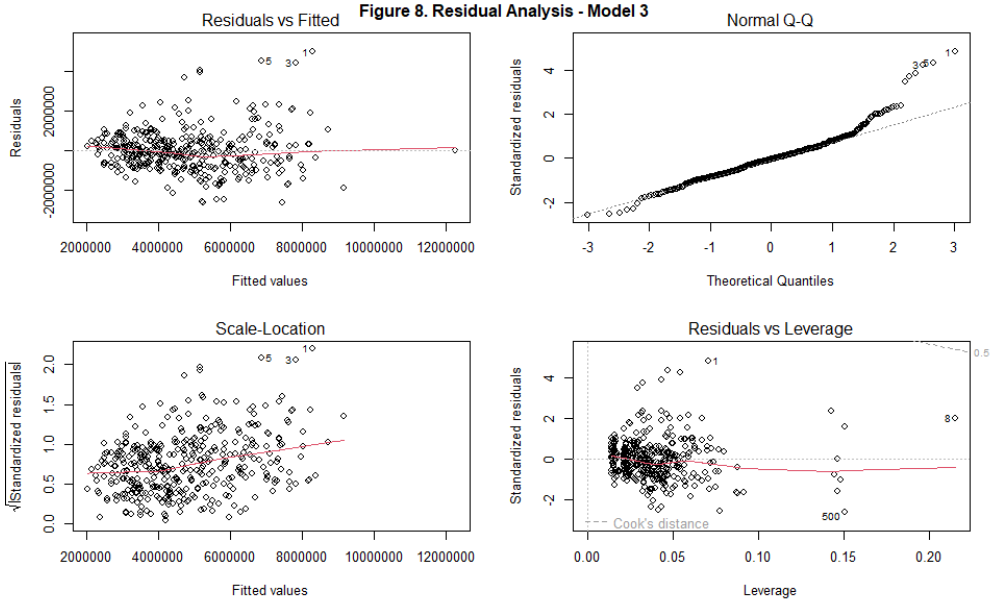


Figure 9. Residual Analysis - Model 5

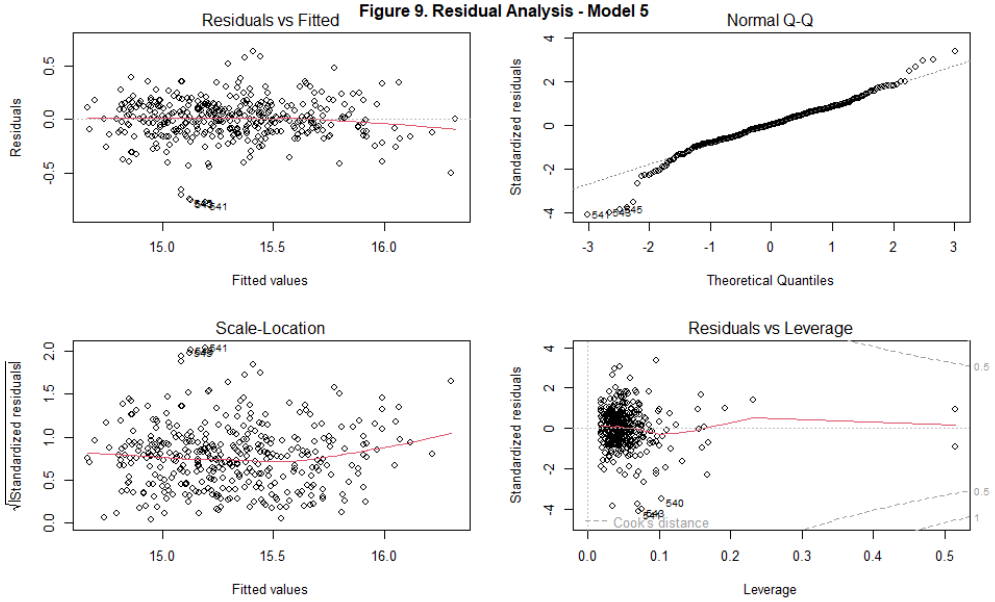


Figure 10. Residual Analysis - Model 7

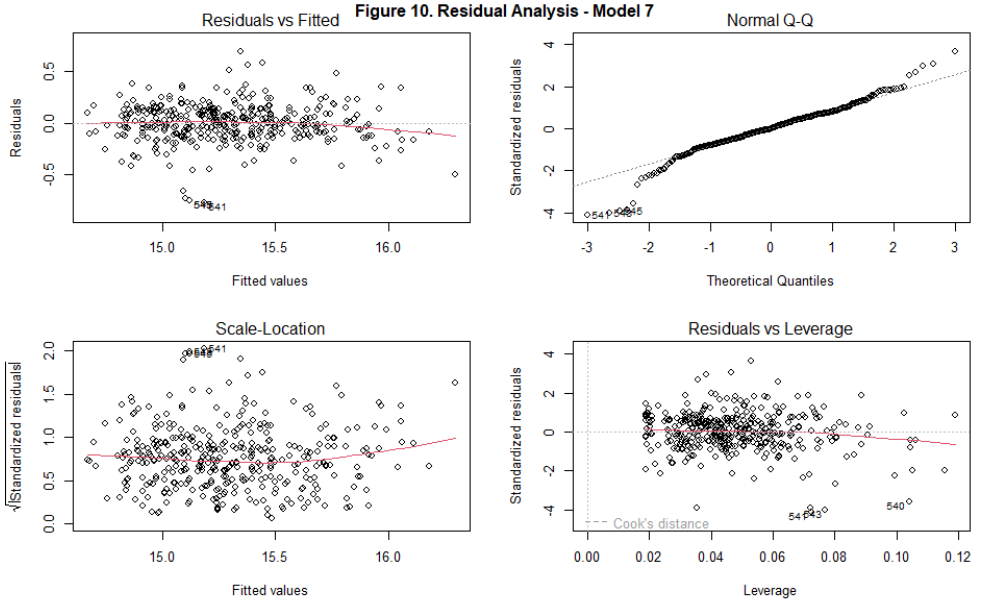


Figure 11. Histogram of Huber Weights

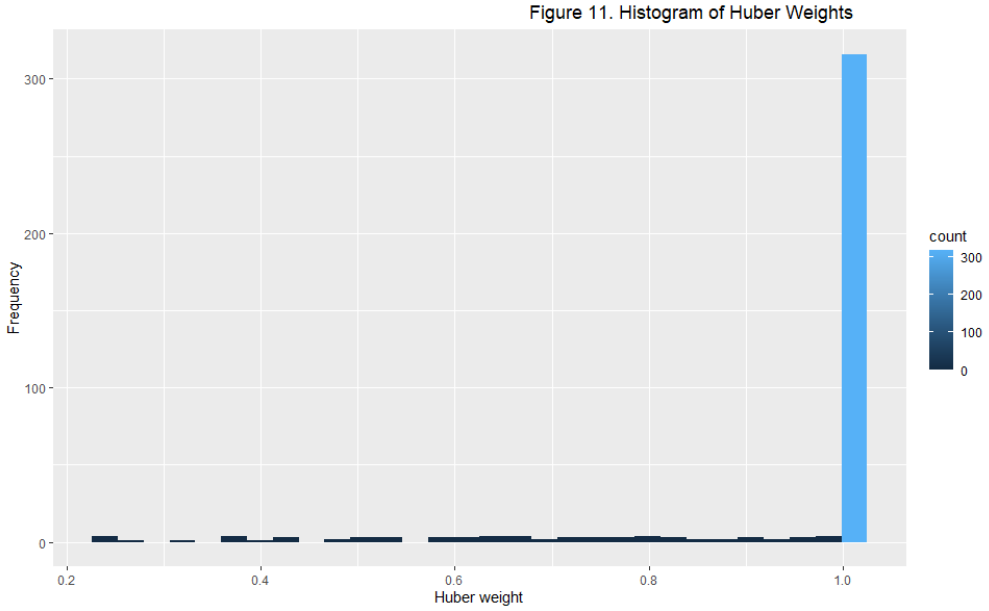


Figure 12. Residual Analysis - Model 9

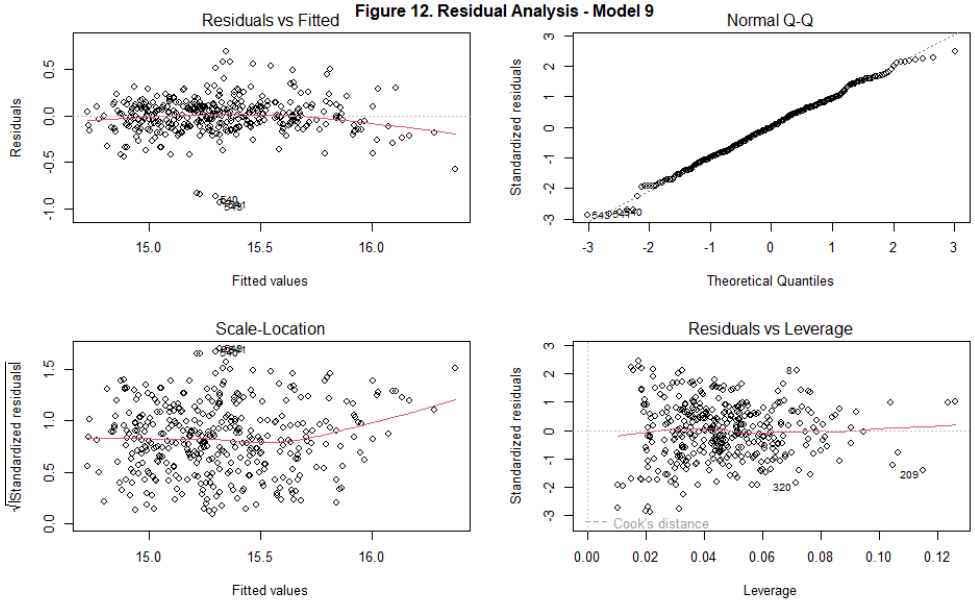


Figure 13. Residual Analysis - Model 10

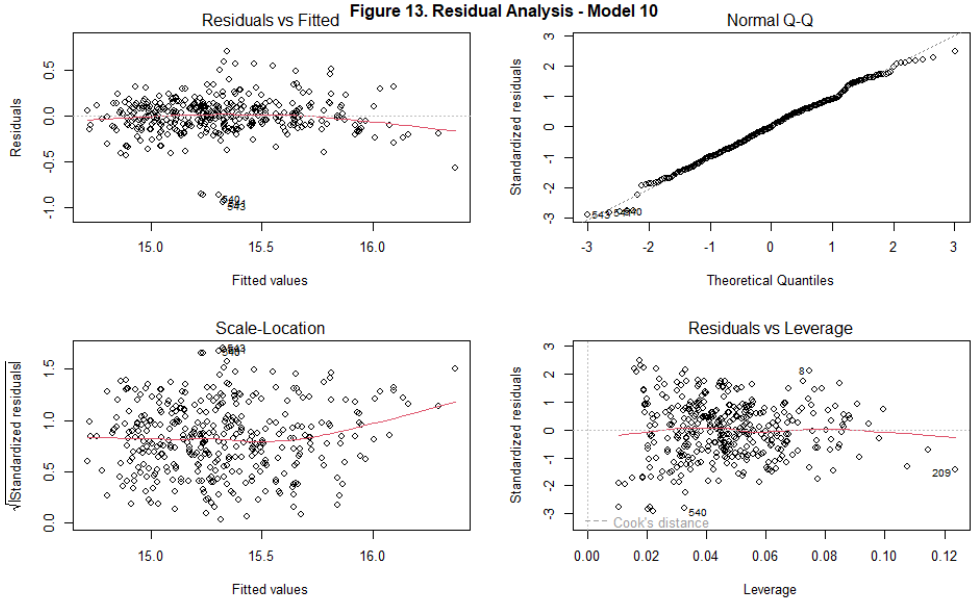
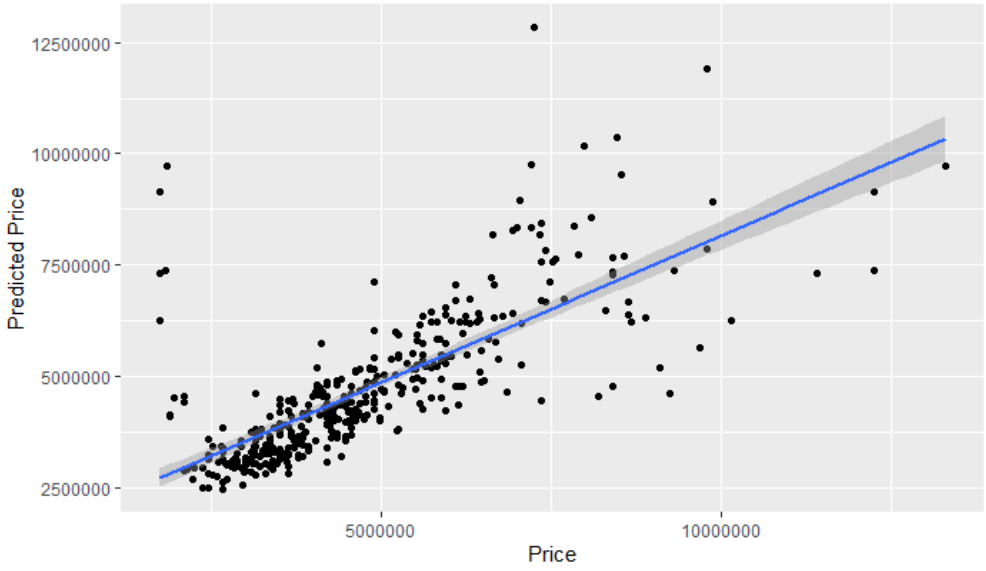


Figure 14. Predicted Price vs Price (Training Data)



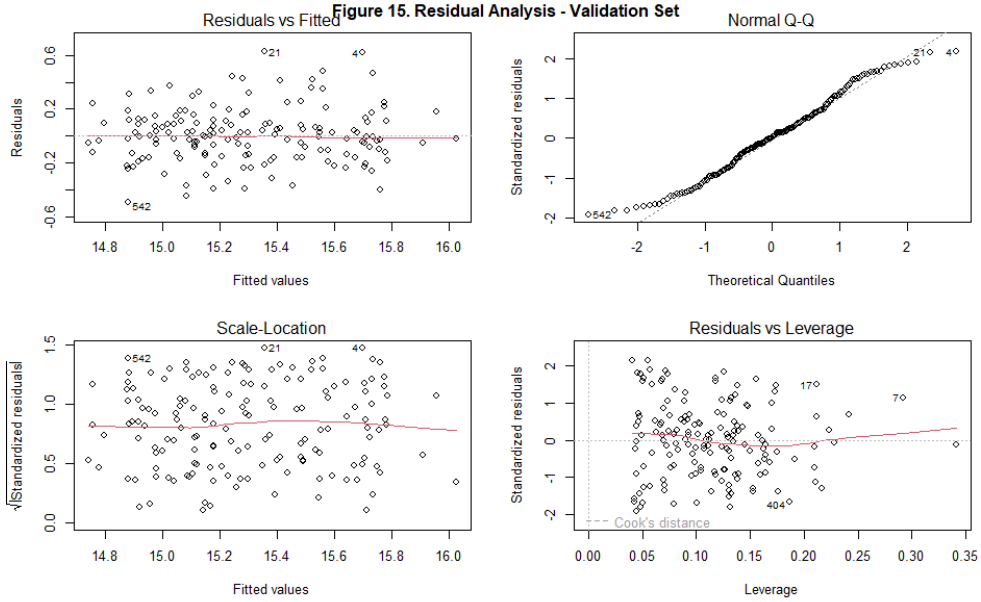


Figure 15. Residual Analysis - Validation Set

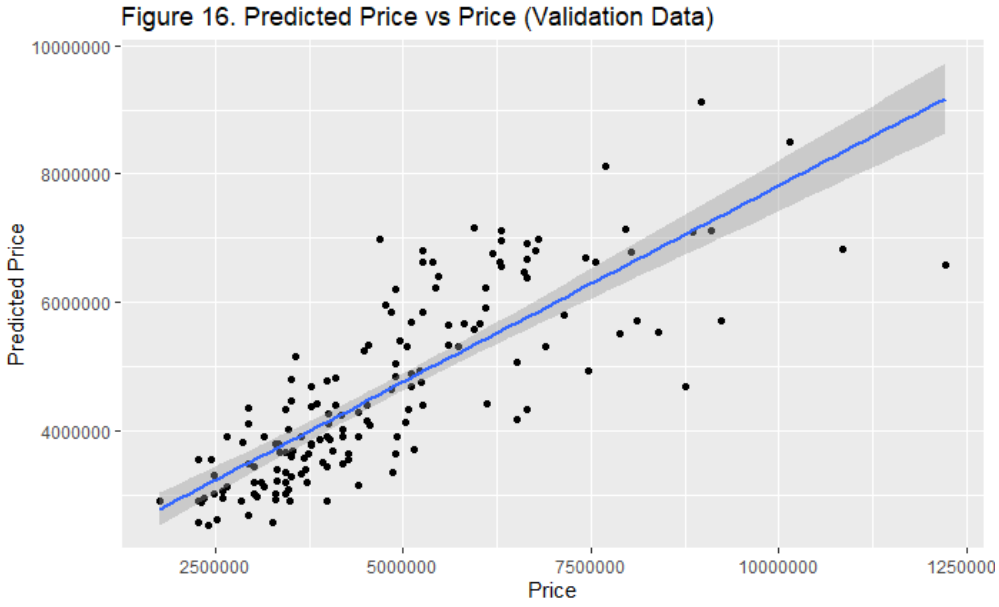


Figure 16. Predicted Price vs Price (Validation Data)

Appendix B. Tables:

Table 1. Training data sample

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking
1	13300000	7420	4	2	3	yes	no	no	no	yes	2
2	12250000	8960	4	4	4	yes	no	no	no	yes	3
3	12250000	9960	3	2	2	yes	no	yes	no	no	2
5	11410000	7420	4	1	2	yes	yes	yes	no	yes	2
8	10150000	16200	5	3	2	yes	no	no	no	no	0
9	9870000	8100	4	1	2	yes	yes	yes	no	yes	2

Table 2. Outliers

	price	area	mainroad	guestroom	basement	hotwaterheating	bed2	bed3	bed4	bed5	bed6plus	bath2	bath3
2	12250000	8960	1	0	0	0	0	0	1	0	0	0	0
8	10150000	16200	1	0	0	0	0	0	0	1	0	0	1
12	9681000	6000	1	1	1	1	0	0	1	0	0	0	1
34	8190000	5960	1	1	1	0	0	1	0	0	0	0	1
67	6930000	13200	1	0	1	1	1	0	0	0	0	0	0
90	6440000	8580	1	0	0	0	0	0	0	1	0	0	1
113	6083000	4300	1	0	0	0	0	0	0	0	1	1	0
144	5600000	4800	0	0	1	1	0	0	0	1	0	1	0
154	5530000	3300	1	0	1	0	0	1	0	0	0	0	1
196	4970000	4410	1	0	1	0	0	0	1	0	0	0	1
291	4200000	2610	0	0	0	0	0	0	1	0	0	0	1

Table 3. Summary of results

#	Train/Validation	Linear/Robust	Full/Step-reduced	Log Transform	Outliers Removed	Huber-Weighted	Adj R-Sqr
1	Train	Linear	Full				0.675
2	Train	Linear	Step				0.677
3	Train	Linear	Step				0.670
4	Train	Linear	Full	Yes			0.723
5	Train	Linear	Step	Yes			0.724
6	Train	Linear	Step	Yes	Yes		0.716
7	Train	Linear	Step	Yes	Yes		0.717
8	Train	Robust	Step	Yes			NA
9	Train	Linear	Step	Yes		Yes	0.774
10	Train	Linear	Step	Yes	Yes	Yes	0.776
11	Validation	Linear	Step	Yes		Yes	0.708

Appendix C. R Code:

```
# load libraries
library(tidyverse)
library(dplyr)
library(corrplot)
library(MASS)
library(dvmisc)
library(car)
library(lmtest)
library(olsrr)
library(caret)
library(kableExtra)

# Load data
house_prices <- read.csv("https://raw.githubusercontent.com/letisalba/Data_621/master/Final_Project/csv

# make this example reproducible
set.seed(1)
```

```r
# use 70% of data set as training set and 30% as evaluation
# set
sample <- sample(c(TRUE, FALSE), nrow(house_prices), replace = TRUE,
    prob = c(0.7, 0.3))
dftrain <- house_prices[sample, ]
dfeval <- house_prices[!sample, ]

head(dftrain) %>%
    kable(caption = "<font color=#000000><b>Table 1.</b> Training data sample</font>") %>%
    kable_styling(bootstrap_options = c("hover", "condensed"),
        font_size = 13)

str(dftrain)

str(dfeval)

summary(dftrain)

summary(dfeval)

# Draw a histogram to figure out the distribution of Sale
# Price
options(scipen = 10000)
ggplot(dftrain, aes(x = price/10^6, fill = ..count..)) + geom_histogram() +
    ggtitle("Figure 1. Histogram of Price") + ylab("Count of houses") +
    xlab("Housing Price in Millions") + theme(plot.title = element_text(hjust = 0.9))

# Draw a histogram to figure out the distribution of Area
options(scipen = 10000)
ggplot(dftrain, aes(x = area, fill = ..count..)) + geom_histogram() +
    ggtitle("Figure 2. Histogram of Area") + ylab("Count of houses") +
    xlab("Area in Square Feet") + theme(plot.title = element_text(hjust = 0.9))

# Counts of categorical variables
dftrain %>%
    count(bedrooms)
dftrain %>%
    count(bathrooms)
dftrain %>%
    count(stories)
dftrain %>%
    count(parking)
dftrain %>%
    count(furnishingstatus)
dftrain %>%
    count(mainroad)
dftrain %>%
    count(guestroom)
dftrain %>%
    count(basement)
dftrain %>%
    count(hotwaterheating)
dftrain %>%
```

```
    count(airconditioning)
dftrain %>%
    count(prefarea)

# Clean function
fun_clean_df <- function(df) {
    df <- df %>%
        mutate(bed2 = ifelse(bedrooms == 2, 1, 0)) %>%
        mutate(bed3 = ifelse(bedrooms == 3, 1, 0)) %>%
        mutate(bed4 = ifelse(bedrooms == 4, 1, 0)) %>%
        mutate(bed5 = ifelse(bedrooms == 5, 1, 0)) %>%
        mutate(bed6plus = ifelse(bedrooms >= 6, 1, 0)) %>%
        dplyr::select(-bedrooms) %>%
        mutate(bath2 = ifelse(bathrooms == 2, 1, 0)) %>%
        mutate(bath3 = ifelse(bathrooms == 3, 1, 0)) %>%
        mutate(bath4plus = ifelse(bathrooms >= 4, 1, 0)) %>%
        dplyr::select(-bathrooms) %>%
        mutate(floor2 = ifelse(stories == 2, 1, 0)) %>%
        mutate(floor3 = ifelse(stories == 3, 1, 0)) %>%
        mutate(floor4plus = ifelse(stories >= 4, 1, 0)) %>%
        dplyr::select(-stories) %>%
        mutate(car1 = ifelse(parking == 1, 1, 0)) %>%
        mutate(car2 = ifelse(parking == 2, 1, 0)) %>%
        mutate(car3plus = ifelse(parking >= 3, 1, 0)) %>%
        dplyr::select(-parking) %>%
        mutate(semifurnished = ifelse(furnishingstatus == "semi-furnished",
            1, 0)) %>%
        mutate(furnished = ifelse(furnishingstatus == "furnished",
            1, 0)) %>%
        dplyr::select(-furnishingstatus) %>%
        mutate(mainroad = ifelse(mainroad == "yes", 1, 0)) %>%
        mutate(guestroom = ifelse(guestroom == "yes", 1, 0)) %>%
        mutate(basement = ifelse(basement == "yes", 1, 0)) %>%
        mutate(hotwaterheating = ifelse(hotwaterheating == "yes",
            1, 0)) %>%
        mutate(ac = ifelse(airconditioning == "yes", 1, 0)) %>%
        dplyr::select(-airconditioning) %>%
        mutate(neighborhood = ifelse(prefarea == "yes", 1, 0)) %>%
        dplyr::select(-prefarea)
}

dftrain_clean <- fun_clean_df(dftrain)
dfeval_clean <- fun_clean_df(dfeval)

# Compare quantitative variables
pairs(dftrain[, c("price", "area", "bedrooms", "bathrooms", "stories",
    "parking")], main = "Figure 3. Comparison of quantiative variables")
# corrplot(cor(dftrain[, c('price', 'area', 'bedrooms',
# 'bathrooms', 'stories', 'parking')], use =
# 'complete.obs'), tl.cex = 0.5)

# Compare nominal categorical variables to price
par(mfrow = c(3, 3))
```

```r
boxplot(price ~ mainroad, data = dftrain)
boxplot(price ~ guestroom, data = dftrain)
boxplot(price ~ basement, data = dftrain)
boxplot(price ~ hotwaterheating, data = dftrain)
boxplot(price ~ airconditioning, data = dftrain)
boxplot(price ~ prefarea, data = dftrain)
boxplot(price ~ furnishingstatus, data = dftrain)
mtext(expression(bold("Figure 4. Comparison of nominal categorical variables")),
    side = 3, line = -1.5, outer = T)

# Compare ordinal categorical variables
par(mfrow = c(2, 2))
boxplot(price ~ bedrooms, data = dftrain)
boxplot(price ~ bathrooms, data = dftrain)
boxplot(price ~ stories, data = dftrain)
boxplot(price ~ parking, data = dftrain)
mtext(expression(bold("Figure 5. Comparison of ordinal categorical variables")),
    side = 3, line = -1.5, outer = T)

# Correlation plot
cor_res <- cor(dftrain_clean)
corrplot(cor_res, type = "lower", order = "original", tl.col = "black",
    tl.srt = 50, tl.cex = 1, main = "Figure 6. Correlation Plot")

# partition data set.seed(10000) train.index <-
# sample(c(1:dim(dftrain_clean)[1]),
# dim(dftrain_clean)[1]*0.8) model_lin_train =
# dftrain_clean[train.index,] model_lin_valid <-
# dftrain_clean[-train.index,]
model_lin_train <- dftrain_clean

# Model 1
lm_mod1 <- lm(price ~ ., data = model_lin_train)
aic_lm_mod1 = AIC(lm_mod1)
summary(lm_mod1)

# Model 2
lm_mod2 <- stepAIC(lm_mod1, trace = F)
aic_lm_mod2 = AIC(lm_mod2)
summary(lm_mod2)

# reduce collinearity, and remove low values
lm_mod3 <- lm(price ~ area + guestroom + basement + bath2 + bath3 +
    bath4plus + floor2 + floor3 + floor4plus + car1 + car2 +
    car3plus + semifurnished + furnished + ac + neighborhood -
    car3plus - bed2, data = model_lin_train)
summary(lm_mod3)

# Define function to calculate mean squared error
calc_mse <- function(lmod) {
    return(mean((summary(lmod))$residuals^2))
}
```

```r
# Define function to aid in model analysis
ModelAnalysis <- function(lmod, plot_title) {

    # Plot residuals
    print("-----------------------------------------------------")
    print(lmod$call)
    par(mfrow = c(2, 2))
    plot(lmod)
    mtext(bquote(bold(.(plot_title))), side = 3, line = -1.5,
        outer = T)
    print("")

    # Shapiro test to determine normality of residuals Null
    # hypothesis: the residuals are normal.  If the p-value
    # is small, reject the null, i.e., consider the
    # residuals *not* normally distributed.
    if (length(lmod$fitted.values) > 3 & length(lmod$fitted.values) <
        5000) {
        st <- shapiro.test(lmod$residuals)
        if (st$p.value <= 0.05) {
            print(paste0("Shapiro test for normality: The p-value of ",
                st$p.value, " is <= 0.05, so reject the null; i.e., the residuals are NOT NORMAL"))
        } else {
            print(paste0("Shapiro test for normality: The p-value of ",
                st$p.value, " is > 0.05, so do not reject the null; i.e., the residuals are NORMAL"))
        }
        print("")
    } else {
        print("Shapiro test for normality of residuals cannot be performed; sample length must be betwe
    }

    # Breusch-Pagan test to determine homoschedasticity of
    # residuals Null hypothesis: the residuals are
    # homoschedastic.  If the p-value is small, reject the
    # null, i.e., consider the residuals heteroschedastic.
    bp <- bptest(lmod)
    if (bp$p.value > 0.05 & bp$statistic < 10) {
        print(paste0("Breusch-Pagan test for homoschedasticity: The p-value of ",
            bp$p.value, " is > 0.05 and the test statistic of ",
            bp$statistic, " is < 10, so don't reject the null; i.e., the residuals are HOMOSCHEDASTIC."
    } else if (bp$p.value <= 0.05) {
        print(paste0("Breusch-Pagan test for homoschedasticity: The p-value of ",
            bp$p.value, " is <= 0.05 and the test statistic is ",
            bp$statistic, ", so reject the null; i.e., the residuals are HETEROSCHEDASTIC."))
    } else {
        print(paste0("Breusch-Pagan test for homoschedasticity: The p-value of ",
            bp$p.value, " and test statistic of ", bp$statistic,
            " are inconclusive, so homoschedasticity can't be determined using this test. But since the
            "it is reasonable to conclude that the residuals are HOMOSCHEDASTIC."))
    }
    print("")

    # Visually look for colinearity - dont do this for
```

```r
# large models pairs(model.matrix(lmod))

# Variance inflation factor (VIF)
print("Variance inflation factor (VIF)")
print("<=1: not correlated, 1-5: moderately correlated, >5: strongly correlated")
print(sort(vif(lmod), decreasing = T))
print("")

# Standardized residual plots (look for points outside
# of 2 or 3 stdev)
p <- length(summary(lmod)$coeff[, 1] - 1)  # number of model parameters
stanres <- rstandard(lmod)
for (i in seq(1, ceiling(p/4))) {
    par(mfrow = c(2, 2))
    starti <- ((i - 1) * 4) + 1
    for (j in seq(starti, starti + 3)) {
        if (j + 1 <= ncol(model.matrix(lmod))) {
            # Skip these plots since we're pretty sure
            # that a linear model isn't valid here
            # plot(model.matrix(lmod)[, j + 1],
            # stanres,
            # xlab=colnames(model.matrix(lmod))[j + 1],
            # ylab='Standardized residuals')
            # abline(h=c(-2, 2), lt=3, col='blue')
            # abline(h=c(-3, 3), lt=2, col='red')
        }
    }
}

# Model scores
print("Model scores:")
print(paste0("    adjusted R-squared: ", round(summary(lmod)$adj.r.squared,
    3)))
print(paste0("    AIC: ", round(AIC(lmod, k = 2), 3)))
print(paste0("    BIC: ", round(BIC(lmod), 3)))
print(paste0("    Mallow's Cp: ", round(ols_mallows_cp(lmod,
    fullmodel = lmod), 3)))
print(paste0("    mean squared error: ", round(calc_mse(lmod),
    3)))
print("")

# Find leverage point cutoff
n <- length(lmod$residuals)
cutoff <- 2 * (p + 1)/n
print(paste0("Leverage point cutoff: ", cutoff))
print("")

# Show points of influence
print("First 10 points of influence:")
poi <- lm.influence(lmod)$hat
len_poi <- length(poi)
ct <- 0
for (i in seq(1, length(poi))) {
```

```
        if (poi[i] > cutoff) {
            ct <- ct + 1
            print(paste0("    case #", i, ": ", round(poi[i],
                3)))
        }
        if (ct > 10) {
            break
        }
    }
    print("")

}

# Analysis on the two step-reduced models
ModelAnalysis(lm_mod2, "Figure 7. Residual Analysis - Model 2")
ModelAnalysis(lm_mod3, "Figure 8. Residual Analysis - Model 3")

# Box-Cox transform on price
bc1 <- powerTransform(price ~ ., data = model_lin_train)
bc1

# Box-Cox result suggests doing a log transform on price
lm_mod4 <- lm(log(price) ~ ., data = model_lin_train)
summary(lm_mod4)
lm_mod5 <- stepAIC(lm_mod4, trace = F)
summary(lm_mod5)
ModelAnalysis(lm_mod5, "Figure 9. Residual Analysis - Model 5")

# Investigate top outliers
model_lin_train[c(2, 5, 9, 25, 49, 62, 77, 103, 110, 136, 210),
    ] %>%
    kable(caption = "<font color=#000000><b>Table 2.</b> Outliers</font>") %>%
    kable_styling(bootstrap_options = c("hover", "condensed"),
        font_size = 13)

# Log transform on price with outliers removed
lm_mod6 <- lm(formula(lm_mod5), data = model_lin_train[c(-2,
    -5, -9, -25, -49, -62, -77, -103, -110, -136, -210), ])
summary(lm_mod6)
lm_mod7 <- stepAIC(lm_mod6, trace = F)
summary(lm_mod7)
ModelAnalysis(lm_mod7, "Figure 10. Residual Analysis - Model 7")

# Huber robust linear regression
lm_mod8 <- rlm(formula(lm_mod7), data = model_lin_train)
dftmp <- data.frame(cbind(price = model_lin_train$price, huber_weight = lm_mod8$w))
dftmp <- dftmp %>%
    arrange(huber_weight, ascending = F)
# hist(dftmp$huber_weight, xlab='Huber weight',
# main='Histogram of Huber Weights')
options(scipen = 10000)
ggplot(dftmp, aes(x = huber_weight, fill = ..count..)) + geom_histogram() +
    ggtitle("Figure 11. Histogram of Huber Weights") + ylab("Frequency") +
```

```
    xlab("Huber weight") + theme(plot.title = element_text(hjust = 0.9))

# New linear model using Huber weights
lm_mod9 <- lm(formula(lm_mod7), weights = lm_mod8$w, data = model_lin_train)
summary(lm_mod9)
ModelAnalysis(lm_mod9, "Figure 12. Residual Analysis - Model 9")

# Remove most significant outliers
lm_mod10 <- lm(formula(lm_mod7), weights = lm_mod8$w[c(-53, -68,
    -87, -90, -103)], data = model_lin_train[c(-53, -68, -87,
    -90, -103), ])
summary(lm_mod10)
ModelAnalysis(lm_mod10, "Figure 13. Residual Analysis - Model 10")

# Five-fold cross validation
set.seed(777)
tc <- trainControl(method = "cv", number = 5)
lmcv <- train(formula(lm_mod8), weights = lm_mod8$w, data = model_lin_train,
    method = "lm", trControl = tc)
print(lmcv)
summary(lmcv)

# Compare predicted price to actual (training data)
model_lin_train$pred_price <- exp(predict(lm_mod10, weights = lm_mod8$w,
    data = model_lin_train, interval = "prediction")[, 1])
model_lin_train %>%
    ggplot(mapping = aes(x = price, y = pred_price)) + geom_point() +
    geom_smooth(method = "lm", se = T) + xlab("Price") + ylab("Predicted Price") +
    ggtitle("Figure 14. Predicted Price vs Price (Training Data)")

# Huber robust linear regression
lm_valid1 <- rlm(formula(lm_mod7), data = dfeval_clean)

# New linear model using Huber weights
lm_valid2 <- lm(formula(lm_mod7), weights = lm_valid1$w, data = dfeval_clean)
summary(lm_valid2)
ModelAnalysis(lm_valid2, "Figure 15. Residual Analysis - Validation Set")

# Compare predicted price to actual (eval data)
dfeval_clean$pred_price <- exp(predict(lm_valid2, weights = lm_valid1$w,
    data = dfeval_clean, interval = "prediction")[, 1])
dfeval_clean %>%
    ggplot(mapping = aes(x = price, y = pred_price)) + geom_point() +
    geom_smooth(method = "lm", se = T) + xlab("Price") + ylab("Predicted Price") +
    ggtitle("Figure 16. Predicted Price vs Price (Validation Data)")

# Model comparison
hdr <- c("#", "Train/Validation", "Linear/Robust", "Full/Step-reduced",
    "Log Transform", "Outliers Removed", "Huber-Weighted", "Adj R-Sqr")
f1 <- c(seq(1:11))
f2 <- c(rep("Train", 10), "Validation")
f3 <- c(rep("Linear", 7), "Robust", rep("Linear", 3))
f4 <- c("Full", "Step", "Step", "Full", rep("Step", 7))
```

```
f5 <- c(rep("", 3), rep("Yes", 8))
f6 <- c(rep("", 5), "Yes", "Yes", "", "", "Yes", "")
f7 <- c(rep("", 8), rep("Yes", 3))
f8 <- round(c(summary(lm_mod1)$adj.r.squared, summary(lm_mod2)$adj.r.squared,
    summary(lm_mod3)$adj.r.squared, summary(lm_mod4)$adj.r.squared,
    summary(lm_mod5)$adj.r.squared, summary(lm_mod6)$adj.r.squared,
    summary(lm_mod7)$adj.r.squared, NA, summary(lm_mod9)$adj.r.squared,
    summary(lm_mod10)$adj.r.squared, summary(lm_valid2)$adj.r.squared),
    3)
dfresult <- data.frame(f1, f2, f3, f4, f5, f6, f7, f8)
colnames(dfresult) <- hdr
dfresult %>%
    kable(caption = "<font color=#000000><b>Table 3.</b> Summary of results</font>") %>%
    kable_styling(bootstrap_options = c("hover", "condensed"),
        font_size = 13)
```