STAT425 Final Project Report

Molin Yang

2025-12-06

Problem Description

People have inelastic demand for housing. Given that different houses have different features—such as the number of bedrooms, bathrooms, construction materials—we sometimes cannot estimate a house's price directly due to the complexity of these features. We can only roughly group houses into a few categories and label as higher -or lower-quality. Hence, it is difficult to verify whether the listed price of a house is reasonable.

The goal of my project is to construct a regression model to estimate an individual house's price based on its known characteristics. The resulting estimate can then be compared with the listing price so that buyers can have a better idea of whether the listing price reflects the true value of the house.

Data Description

In this project I want to focus on the housing market in a large city. Because the estimation relies on past sales data, I selected a dataset with large observations to stabilize the predictors and make the model's prediction results more convincing. I also preferred a dataset with a large number of variables so that I can fit a full model and select useful predictors. I found a suitable dataset on Kaggle (Harlfoxem, 2016), which includes 21,613 housing sales recorded between May 2014 and May 2015 in King County, USA, where Seattle is located, along with 20 variables.

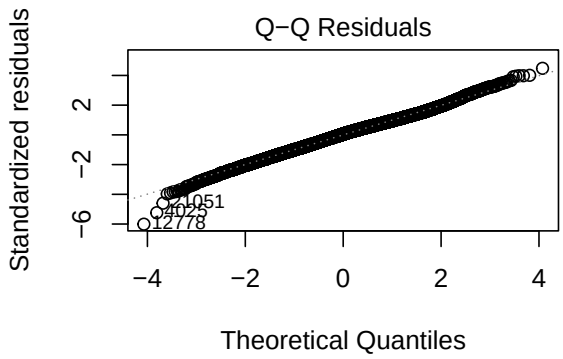
The response variable is the house sale price. The variables include the number of bedrooms, number of bathrooms, size of the above-ground area (sqft), land size (sqft), livingroom size (sqft), number of floors, overall grade of property (ranging from 1 to 13, based on construction material and architectural design quality), view score (ranging from 1 to 5), and year it was built. There are some irrelevant features such as the longitude, latitude, and the sale date of the house.

Most predictors are numerical (discrete or continuous), and the rest are ordinal or categorical. Before modeling, I cleaned the data by removing irrelevant variables. I also omitted observations with missing values.

Methodology and Statistical Approach

Multiple Linear Regression (OLS Method): I first fitted a multiple linear regression model using the OLS method. I selected number of bedrooms, number of bathrooms, livingroom size, land size, above-ground area size, years built, number of floors, waterfront, view, condition, and grade as predictors. I then used the mean shift test with 21601 degrees of freedom to detect outliers and removed these observations from the dataset. This ensures that the regression results are not affected by extreme values and can perform better in estimating housing prices.

The next step is to conduct model diagnostics. The first assumption I checked is the linearity assumption. I used the boxcox function in the MASS library to find the optimal λ value for this model. The result is $\lambda_{opt} \approx 0.1$. Since this value is close to zero, a log transformation is preferred. I applied log function to the housing prices and fitted the model again using the log-transformed housing prices as the response variable, and the linearity problem was addressed.



The second assumption I checked is the normality assumption of the error term. Displayed above is the Q-Q plot. The points fall roughly along a straight line, indicating that the normality assumption is satisfied.

The third assumption I checked is the constant variance assumption of the error term. I performed the BP test, and found the p-value was less than 0.05, indicating the presence of heteroscedasticity problem, which needs to be addressed with a better model.

The fourth assumption I checked is whether multicollinearity exists. I first constructed the X matrix, then centered and standardized the predictors, computed eigenvalues of X^TX, and used them to calculate the k-values. Since the largest k-value is 7.6296, which is less than 30, multicollinearity is not a concern.

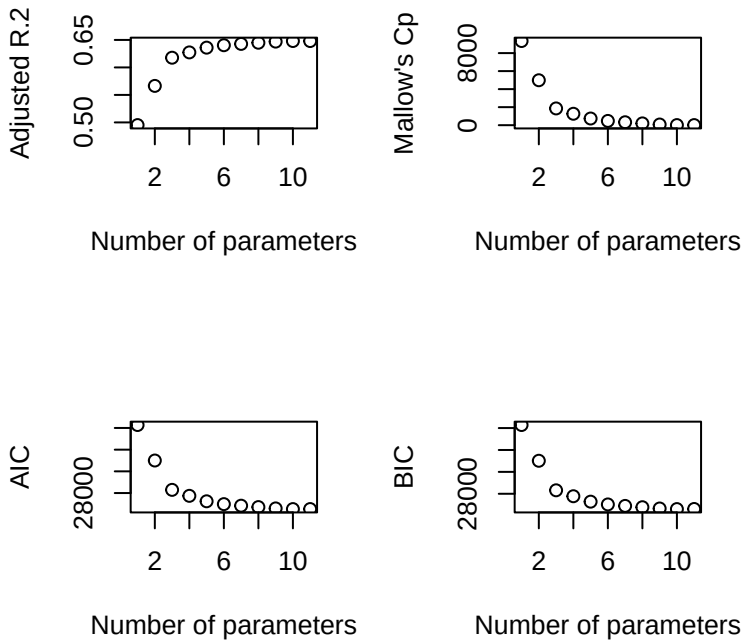
Before moving on to build a better model, I will present the results of this multiple linear regression model using the OLS method:

$$\begin{aligned}\widehat{\log(\text{price})} = 21.23 &- 0.02269 N_{beds} + 0.07634 N_{baths} + 2.116\times10^{-4} S_{living} \\ &- 2.039\times10^{-8}\, S_{land} - 6.308\times10^{-5}\, S_{above} - 5.40\times10^{-3}\, T_{built} \\ &+ 0.1048\, N_{floors} + 0.3350\,\text{Water} + 0.04722\,\text{View} \\ &+ 0.04080\,\text{condition} + 0.2277\,\text{grade}.\end{aligned}$$

This model has $R^2_{adj} = 0.6467$ and all predictors are significant except land size.

Multiple Linear Regression (WLS Method) and Variable Selection: Since the multiple linear model using the OLS method faced heteroscedasticity problem, the model is still unbiased, but its standard errors become unreliable. This can result in inefficient OLS estimators and less accurate results in high-variance regions. To address this problem, I used the WLS method to construct my second model with the same predictor variables. I modeled the log-squared residuals from the OLS model to estimate how the error variance changes with predictors. The predicted variances were then exponentiated and inverted to obtain the WLS weights. $w_i = \frac{1}{\hat{\sigma}_i^2}$. Therefore, observations with larger variance or larger OLS residuals receive smaller weights in this new model. Model diagnostics were applied to the second model, and all the model assumptions were satisfied. Therefore, we can perform variable selection and find the best model.

I used the leaps and bounds method to select variables. By graphing out the 11 model candidates' R^2_{adj}, Mallow's Cp, AIC, and BIC (detailed values are provided in the appendix), It is clear that the model with 10 variables has the smallest Mallow's Cp, AIC, and BIC values. It has the same R^2_{adj} as the full model:



We conclude that this model is the best model, and the variable omitted is land size.

I fitted the model again with the remaining 10 variables. The resulting model is:

$$\begin{aligned}\widehat{\log(\text{price})} = 20.96 &- 0.02367 N_{beds} + 0.07121 N_{baths} + 2.192\times10^{-4} S_{living} - 7.258\times10^{-5}\, S_{above} \\ &- 5.262\times10^{-3}\, T_{built} + 0.148\, N_{floors} + 0.3316\,\text{Water} + 0.04621\,\text{View} + 0.04218\,\text{condition} + 0.2261\,\text{grade}.\end{aligned}$$

This model has $R^2_{adj} = 0.648$, a slight increase compared to model 1. However, this model successfully addressed heteroscedasticity problem and all predictors are significant, so model 2 is more desirable than model 1.

Interpretation and Insight

The WLS model after variable selection shows that bathrooms, floors, water proximity, and overall construction and architectural design quality are stronger predictors of housing prices in Seattle. The log transformation and WLS method have improved the reliability of the inference. This model can therefore help potential buyers to better understand the true value of a property. A comparison between predicted and listed prices is possible, enabling buyers to discern mispricing.

However, this model still has limitations. The major problem is the data are outdated. Since the sales were recorded about a decade ago, the results may not precisely reflect today's market. Although adjusting the intercept could correct the overall inflation, a desirable dataset for today's Seattle house sales is not available. In addition, the model is limited to predicting Seattle housing prices and cannot generalize to other regions because housing market vary across states.

Citations and Ethics

Harlfoxem. (2016). House Sales in King County, USA [Data set]. Kaggle.

https://www.kaggle.com/datasets/harlfoxem/housesalesprediction

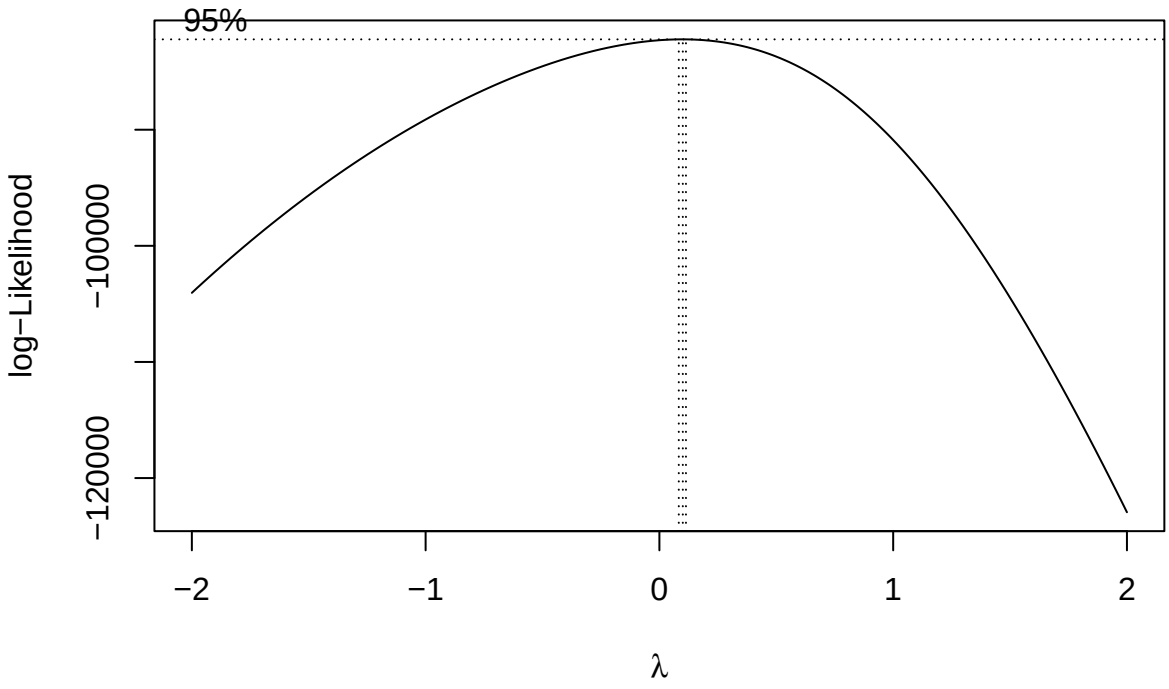
Appendix

```r
## load data
dat = read.csv("kc_house_data.csv", header = TRUE)

## remove redundant columns and remove NA values
dat = dat[ , -c(1, 2, 16:21)]
dat = na.omit(dat)

## check the length of each column is the same after data cleaning
sapply(dat, length) ## all columns have 21613 non-NA entries -> we can proceed
```

```
##         price      bedrooms     bathrooms   sqft_living      sqft_lot
##         21613         21613         21613         21613         21613
##        floors    waterfront          view     condition         grade
##         21613         21613         21613         21613         21613
##    sqft_above sqft_basement      yr_built
##         21613         21613         21613
```

```r
library(MASS)
library(lmtest)


## fit original full model
fullmodel = lm(price ~ bedrooms + bathrooms +
            sqft_living + sqft_lot + sqft_above +
            yr_built + floors + waterfront + view + condition + grade,
            data = dat)



## identify and remove outliers using mean shift test
## (df = n-p-1, p: # of predictors excluding intercept)

n = dim(dat)[1]
tvals = sort(abs(rstudent(fullmodel)), decreasing = TRUE)
## we might need to clean 107 obs as outliers

mean.shift.test.threshold = abs(qt(0.05/(n * 2), 21601))

outliers = which(abs(tvals) > mean.shift.test.threshold)
dat = dat[-outliers, ]
```

```r
## linearity check
sr.trans = boxcox(fullmodel, lambda = seq(-2, 2, length = 500))
```

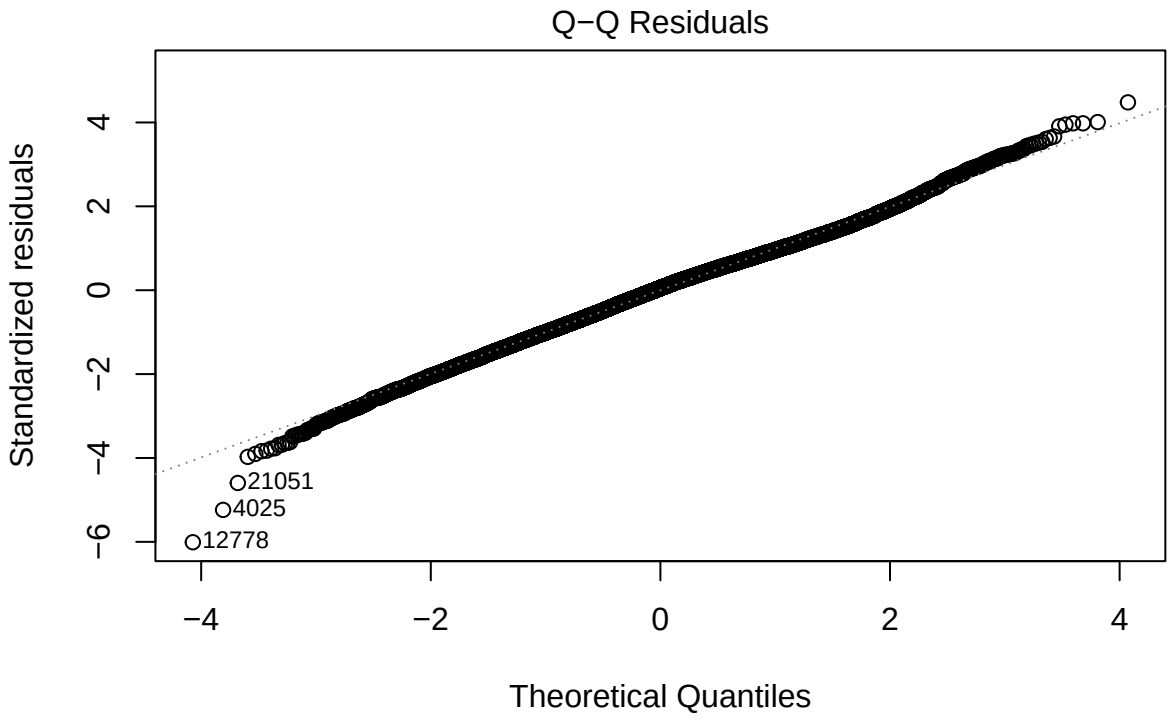



```
sr.trans$x[sr.trans$y == max(sr.trans$y)] ## need log transformation
```

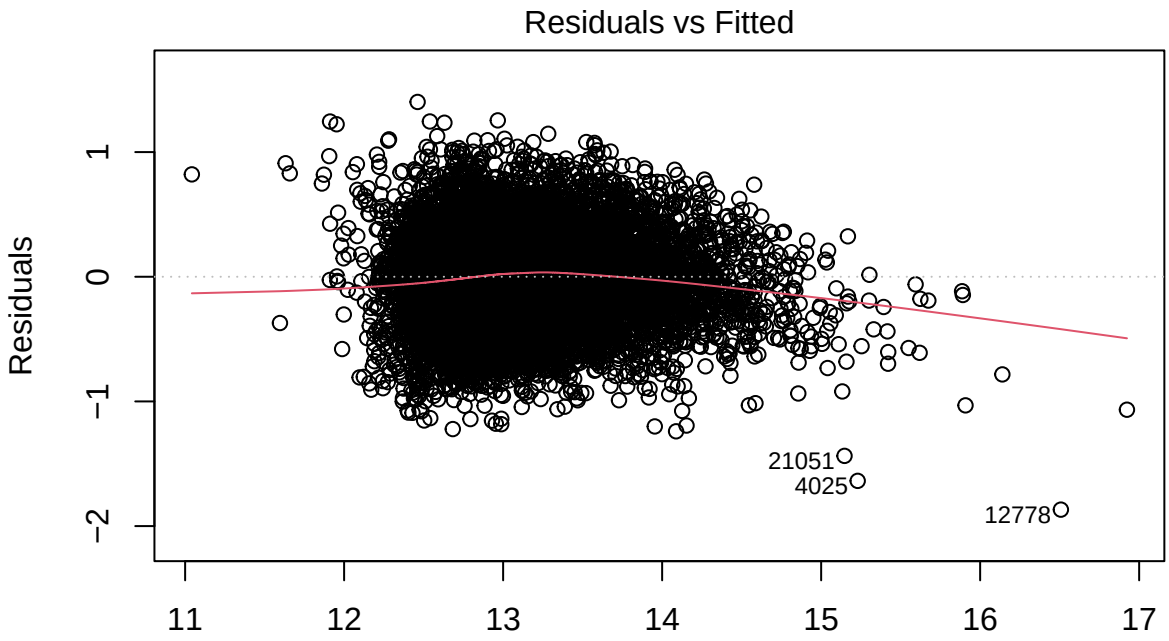
```
## [1] 0.1002004
```

```
## log transformation to response variable
dat$log_price = log(dat$price)
dat = dat[ , -1]
## fit the full model again using log(price) as response variable
fullmodel.log = lm(log_price ~ bedrooms + bathrooms +
            sqft_living + sqft_lot + sqft_above +
            yr_built + floors + waterfront + view + condition + grade, data = dat)
```

```
## check normality assumption of the error term
plot(fullmodel.log, which = 2, sub = "") ## normality assumption of error term satisfied
```




```
## check constant variance assumption of the error term
plot(fullmodel.log, which = 1)
```




```
bptest(fullmodel.log) ## heteroscedasticity problem exists and needed to be improved
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fullmodel.log
## BP = 428.9, df = 11, p-value < 2.2e-16
```

```
## multicollinearity check
x = model.matrix(fullmodel.log)
x = x[ , -1]
x = x - matrix(apply(x, 2, mean), 21506, 11, byrow = TRUE)
x = x / matrix(apply(x, 2, sd), 21506, 11, byrow = TRUE)
e = eigen(t(x) %*% x)
sqrt(e$val[1] / e$val)
```

```
##  [1] 1.000000 1.665528 1.888193 2.080230 2.487996 2.641403 2.913300 2.952611
##  [9] 3.875961 4.291151 7.629564
```

```
## k-values all less than 30. Multicollinearity problem is not a concern

summary(fullmodel.log)
```

```
##
## Call:
## lm(formula = log_price ~ bedrooms + bathrooms + sqft_living +
##     sqft_lot + sqft_above + yr_built + floors + waterfront +
##     view + condition + grade, data = dat)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.86805 -0.21091  0.01593  0.20971  1.40218
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.123e+01  1.895e-01 112.002  < 2e-16 ***
## bedrooms    -2.269e-02  2.938e-03  -7.723 1.19e-14 ***
## bathrooms    7.634e-02  5.021e-03  15.205  < 2e-16 ***
## sqft_living  2.116e-04  6.676e-06  31.703  < 2e-16 ***
## sqft_lot    -2.039e-08  5.299e-08  -0.385      0.7
## sqft_above  -6.308e-05  6.462e-06  -9.761  < 2e-16 ***
## yr_built    -5.400e-03  9.742e-05 -55.431  < 2e-16 ***
## floors       1.048e-01  5.413e-03  19.367  < 2e-16 ***
## waterfront   3.350e-01  2.704e-02  12.392  < 2e-16 ***
## view         4.722e-02  3.273e-03  14.426  < 2e-16 ***
## condition    4.080e-02  3.579e-03  11.401  < 2e-16 ***
## grade        2.277e-01  3.140e-03  72.523  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3131 on 21494 degrees of freedom
## Multiple R-squared:  0.6469, Adjusted R-squared:  0.6467
## F-statistic:  3579 on 11 and 21494 DF,  p-value: < 2.2e-16
```

```
## build weight
e2 <- residuals(fullmodel.log)^2
var.model <- lm(log(e2) ~ bedrooms + bathrooms + sqft_living + sqft_lot +
                sqft_above + yr_built + floors +
                waterfront + view + condition + grade,
                data = dat)
pred.var <- exp(predict(var.model))
w <- 1 / pred.var



## fit full model using WLS method to address heteroscedasticity problem
wls.model = lm(log_price ~ bedrooms + bathrooms +
            sqft_living + sqft_lot + sqft_above +
            yr_built + floors + waterfront + view + condition + grade,
            data = dat, weights = w)


## multicollinearity check
x = model.matrix(wls.model)
x = x[ , -1]
x = x - matrix(apply(x, 2, mean), 21506, 11, byrow = TRUE)
x = x / matrix(apply(x, 2, sd), 21506, 11, byrow = TRUE)
e = eigen(t(x) %*% x)
sqrt(e$val[1] / e$val)
```

```
##  [1] 1.000000 1.665528 1.888193 2.080230 2.487996 2.641403 2.913300 2.952611
##  [9] 3.875961 4.291151 7.629564
```

```r
## k-values all less than 30. Multicollinearity problem is not a concern

summary(wls.model)
```

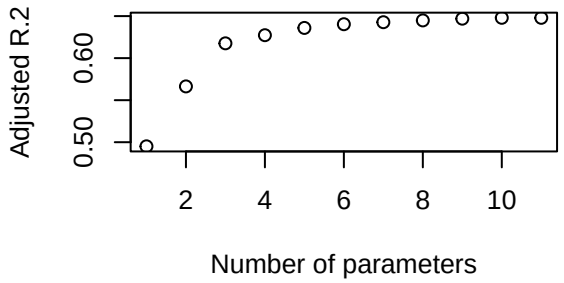
```
##
## Call:
## lm(formula = log_price ~ bedrooms + bathrooms + sqft_living +
##     sqft_lot + sqft_above + yr_built + floors + waterfront +
##     view + condition + grade, data = dat, weights = w)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8442 -1.2617  0.0966  1.2561  7.7693
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.096e+01  1.870e-01 112.100  < 2e-16 ***
## bedrooms    -2.371e-02  2.975e-03  -7.968 1.69e-15 ***
## bathrooms    7.120e-02  4.894e-03  14.546  < 2e-16 ***
## sqft_living  2.192e-04  6.640e-06  33.015  < 2e-16 ***
## sqft_lot    -7.686e-09  4.611e-08  -0.167    0.868
## sqft_above  -7.247e-05  6.333e-06 -11.442  < 2e-16 ***
## yr_built    -5.261e-03  9.567e-05 -54.992  < 2e-16 ***
## floors       1.147e-01  4.940e-03  23.227  < 2e-16 ***
## waterfront   3.315e-01  2.985e-02  11.108  < 2e-16 ***
## view         4.624e-02  3.263e-03  14.170  < 2e-16 ***
## condition    4.218e-02  3.626e-03  11.634  < 2e-16 ***
## grade        2.261e-01  3.072e-03  73.612  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.852 on 21494 degrees of freedom
## Multiple R-squared:  0.6481, Adjusted R-squared:  0.648
## F-statistic:  3599 on 11 and 21494 DF,  p-value: < 2.2e-16
```

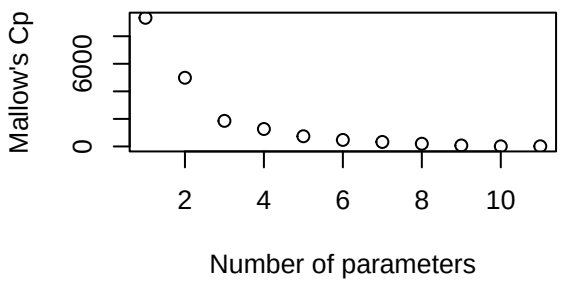
```r
## we want to use leaps and bounds method to see whether we can produce a more concise model
library(leaps)


l.and.b = regsubsets(log_price ~ bedrooms + bathrooms +
            sqft_living + sqft_lot + sqft_above +
            yr_built + floors + waterfront + view + condition + grade,
            data = dat, weights = w, nvmax = 11)

rs = summary(l.and.b)
rs$which
```

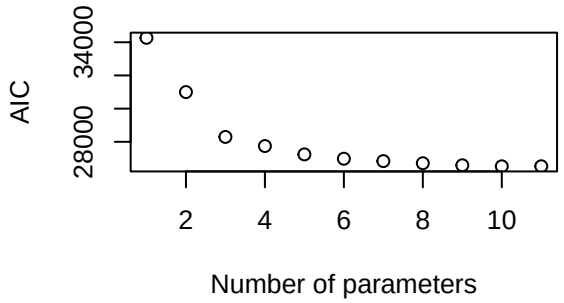
```
##    (Intercept) bedrooms bathrooms sqft_living sqft_lot sqft_above yr_built
## 1         TRUE    FALSE     FALSE       FALSE    FALSE      FALSE    FALSE
## 2         TRUE    FALSE     FALSE       FALSE    FALSE      FALSE     TRUE
## 3         TRUE    FALSE     FALSE        TRUE    FALSE      FALSE     TRUE
## 4         TRUE    FALSE     FALSE        TRUE    FALSE      FALSE     TRUE
## 5         TRUE    FALSE     FALSE        TRUE    FALSE      FALSE     TRUE
## 6         TRUE    FALSE      TRUE        TRUE    FALSE      FALSE     TRUE
## 7         TRUE    FALSE      TRUE        TRUE    FALSE      FALSE     TRUE
```

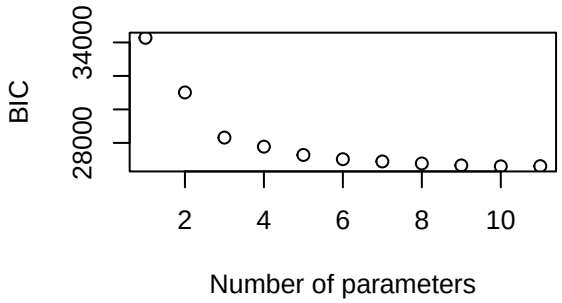
```
## 8         TRUE    FALSE      TRUE        TRUE    FALSE       TRUE     TRUE
## 9         TRUE    FALSE      TRUE        TRUE    FALSE       TRUE     TRUE
## 10        TRUE     TRUE      TRUE        TRUE    FALSE       TRUE     TRUE
## 11        TRUE     TRUE      TRUE        TRUE     TRUE       TRUE     TRUE
##    floors waterfront  view condition grade
## 1   FALSE      FALSE FALSE     FALSE  TRUE
## 2   FALSE      FALSE FALSE     FALSE  TRUE
## 3   FALSE      FALSE FALSE     FALSE  TRUE
## 4    TRUE      FALSE FALSE     FALSE  TRUE
## 5    TRUE      FALSE  TRUE     FALSE  TRUE
## 6    TRUE      FALSE  TRUE     FALSE  TRUE
## 7    TRUE      FALSE  TRUE      TRUE  TRUE
## 8    TRUE       TRUE  TRUE     FALSE  TRUE
## 9    TRUE       TRUE  TRUE      TRUE  TRUE
## 10   TRUE       TRUE  TRUE      TRUE  TRUE
## 11   TRUE       TRUE  TRUE      TRUE  TRUE
```

```r
n = dim(dat)[1]
msize = 1:11

Aic = n * log(rs$rss/n) + 2*msize
Bic = n * log(rs$rss/n) + msize*log(n)

par(mfrow = c(2, 2))
plot(msize, rs$adjr2, xlab = "Number of parameters", ylab = "Adjusted R^2")
plot(msize, rs$cp, xlab = "Number of parameters", ylab = "Mallow's Cp")
plot(msize, Aic, xlab = "Number of parameters", ylab = "AIC")
plot(msize, Bic, xlab = "Number of parameters", ylab = "BIC")
```










```
model_sizes = c(9, 10, 11)

df_summary = data.frame(
  ModelSize = model_sizes,
  AdjR2 = rs$adjr2[model_sizes],
  Cp = rs$cp[model_sizes],
  AIC = Aic[model_sizes],
  BIC = Bic[model_sizes]
)

df_summary
```

```
##   ModelSize     AdjR2       Cp      AIC      BIC
## 1         9 0.6469476 71.70414 26584.03 26655.81
## 2        10 0.6479741 10.02779 26522.41 26602.17
## 3        11 0.6479582 12.00000 26524.38 26612.12
```

```
wls.model.new = lm(log_price ~ bedrooms + bathrooms +
            sqft_living + sqft_above +
            yr_built + floors + waterfront + view + condition + grade,
            data = dat, weights = w)
summary(wls.model.new)
```

```
##
## Call:
## lm(formula = log_price ~ bedrooms + bathrooms + sqft_living +
##     sqft_above + yr_built + floors + waterfront + view + condition +
```

```
##     grade, data = dat, weights = w)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -9.8512 -1.2617  0.0968  1.2560  7.7705
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.096e+01  1.869e-01  112.12  < 2e-16 ***
## bedrooms    -2.367e-02  2.966e-03   -7.98 1.54e-15 ***
## bathrooms    7.121e-02  4.894e-03   14.55  < 2e-16 ***
## sqft_living  2.192e-04  6.636e-06   33.03  < 2e-16 ***
## sqft_above  -7.258e-05  6.299e-06  -11.52  < 2e-16 ***
## yr_built    -5.262e-03  9.564e-05  -55.02  < 2e-16 ***
## floors       1.148e-01  4.919e-03   23.34  < 2e-16 ***
## waterfront   3.316e-01  2.984e-02   11.11  < 2e-16 ***
## view         4.621e-02  3.260e-03   14.18  < 2e-16 ***
## condition    4.218e-02  3.626e-03   11.63  < 2e-16 ***
## grade        2.261e-01  3.070e-03   73.67  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.852 on 21495 degrees of freedom
## Multiple R-squared:  0.6481, Adjusted R-squared:  0.648
## F-statistic:  3959 on 10 and 21495 DF,  p-value: < 2.2e-16
```