Step4 Shrinkage method and weighted least square regression

Liuqian Bao, Stella Ramirez, Andrew Cheng

2023-10-27

Introduction:

In this step, we will use data we have used for our previous steps and incorporate the shrinkage methods.

The citation for our original data source is: Anna Montoya, DataCanary. (2016).House Prices - Advanced Regression Techniques. Kaggle.Obtained from https://kaggle.com/competitions/house-prices-advanced-regression-techniques (https://kaggle.com/competitions/house-prices^advanced-regression-techniques)

The population that we are inferring our results on are all residential houses in Ames, Iowa.

For step 4 of our project, we will execute ridge regression and LASSO on our dataset. First, we will investigate whether there may be collinearity issues. In order to reduce the errors this collinearity will cause, we need to shrink or even remove the irrelevant predictors.

To detect collinearity, we created a pairwise correlation chart. Variables that have high correlation will have values closer to positive or negative 1.

```
##             LotArea GarageCars BsmtFinSF1 FullBath OpenPorchSF HalfBath
## LotArea        1.00       0.15       0.21     0.13        0.08     0.01
## GarageCars     0.15       1.00       0.22     0.47        0.21     0.22
## BsmtFinSF1     0.21       0.22       1.00     0.06        0.11     0.00
## FullBath       0.13       0.47       0.06     1.00        0.26     0.14
## OpenPorchSF    0.08       0.21       0.11     0.26        1.00     0.20
## HalfBath       0.01       0.22       0.00     0.14        0.20     1.00
```

As you can see, a few of our variables have some correlation. While there aren't too many with extremely high correlation, it can still be beneficial to perform ridge and lasso regression in order to use a model that only uses the useful information. The goal is that, after standardization, the coefficient estimates will have a smaller variance.

Ridge Regression

Ridge Regression is a technique used that keeps all variables, but still helps to filter irrelevant information. We will be shrinking the size of the coefficients. In order to use the glmnet package, we need to ensure that our response is a vector of log(SalePrice), and our predictors are in the form of a data.matrix.

Fit Ridge Regression Model

Next, we use the glmnet package to fit a model for ridge regression. The glmnet function automatically standardizes our predictor variables. This standardization makes it so that the standard deviation of each variable is 1 and then mean is 0.

```
##           Length Class     Mode
## a0        100    -none-    numeric
## beta      800    dgCMatrix S4
## df        100    -none-    numeric
## dim         2    -none-    numeric
## lambda    100    -none-    numeric
## dev.ratio 100    -none-    numeric
## nulldev     1    -none-    numeric
## npasses     1    -none-    numeric
## jerr        1    -none-    numeric
## offset      1    -none-    logical
## call        4    -none-    call
## nobs        1    -none-    numeric
```

Finding Optimal Lambda

Next, we use cross validation to find the best lambda value. In order to do this process, 1/10 of the data is tested at a time.

```
## [1] 0.02717837
```

The optimal lambda found by cross-validation is 0.027.

The following shows a plot of the test MSE with the $log(\lambda)$ values.

Next, we view the estimates of the coefficients for the best model.

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept)  1.180188e+01
## LotArea      4.206529e-06
## GarageCars   1.682761e-01
## BsmtFinSF1   1.636289e-04
## FullBath     1.885228e-01
## ExterQual   -8.341086e-02
## OpenPorchSF  3.078674e-04
## HalfBath     1.135287e-01
## KitchenQual -7.475280e-02
```

So that we could explore ridge regression further, and view what happens as lambda increases, we created the following graph.

Finally, we wanted to see what the r^2 value is for our ridge regression model.

```
## [1] 0.7132206
```

This value of r^2 means that the best model possible, found as a result of ridge regression, explains 71.32% of the variation in the response.

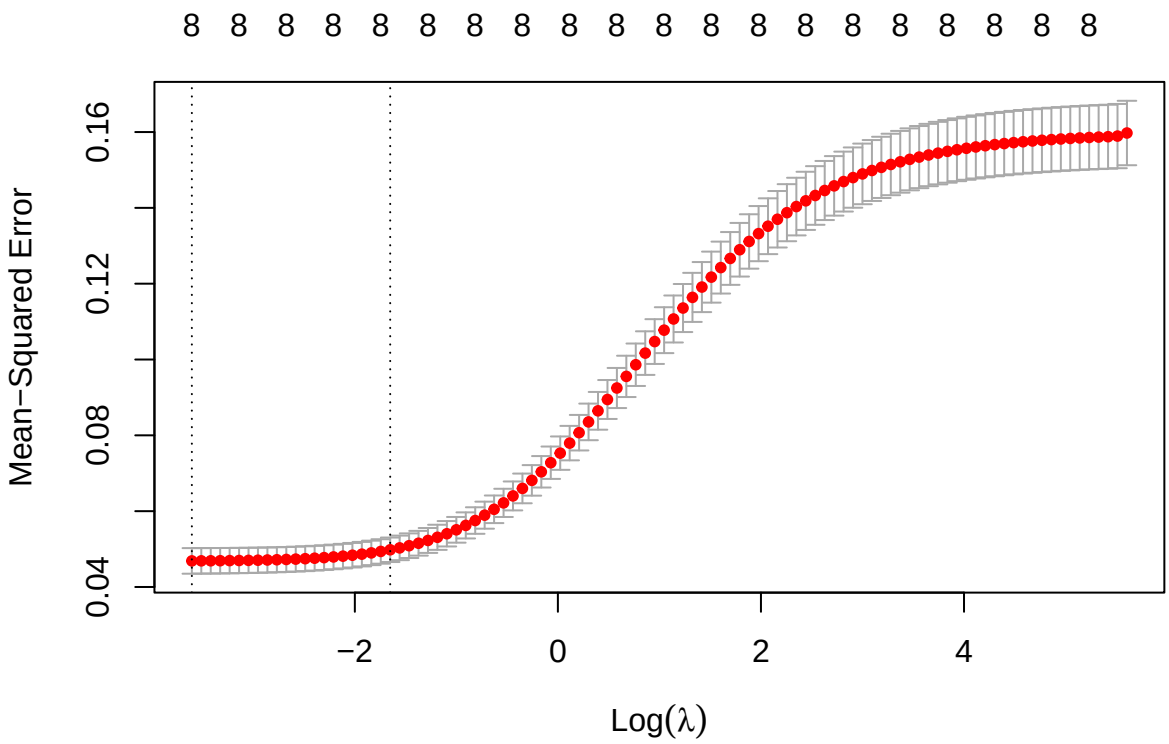

Figure 1: MSE and log(lambda)(Ridge model)

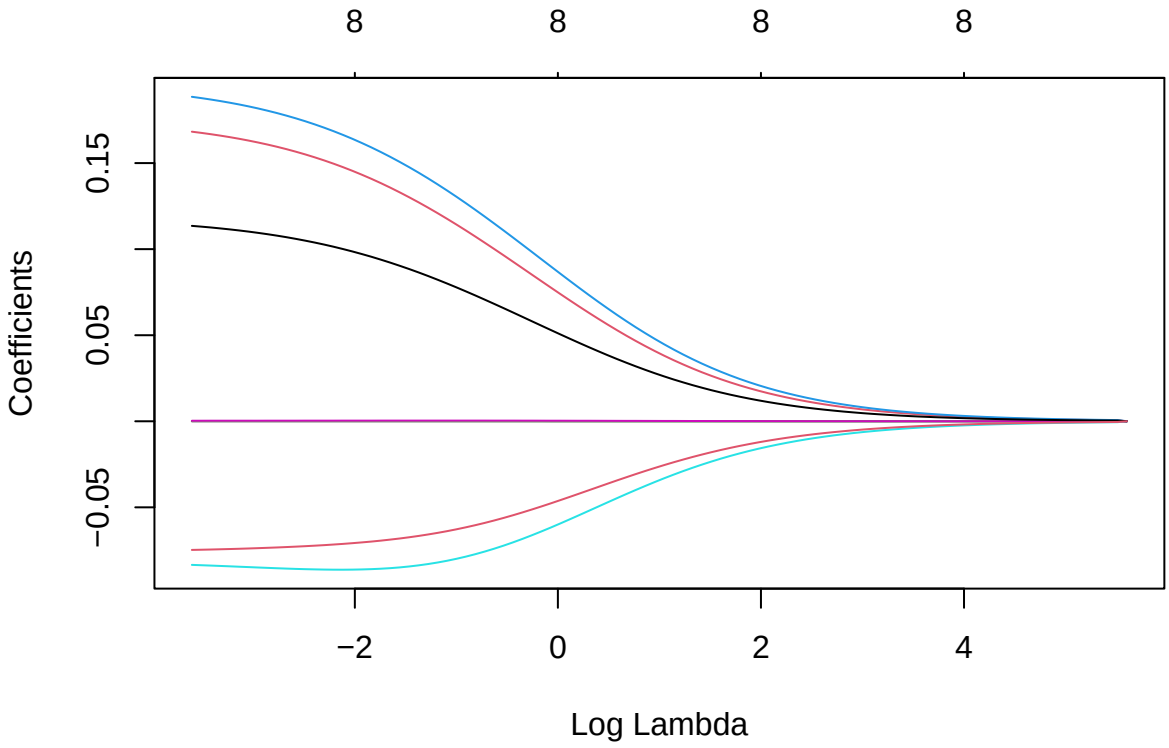

Figure 2: Effect on coefficients as Lambda Increases(Ridge model)

Lasso Regression

We executed Lasso regression on the complete data set with the final MLR model from the previous project task. With Lasso Regression, some of the β's are shrunk all the way to zero. This will mean that the corresponding irrelevant predictor will not have influence in our model.

Fit Lasso Model

The following is a summary of our fitted lasso model.

```
##           Length Class     Mode
## a0         60    -none-    numeric
## beta      480    dgCMatrix S4
## df         60    -none-    numeric
## dim         2    -none-    numeric
## lambda     60    -none-    numeric
## dev.ratio  60    -none-    numeric
## nulldev     1    -none-    numeric
## npasses     1    -none-    numeric
## jerr        1    -none-    numeric
## offset      1    -none-    logical
## call        4    -none-    call
## nobs        1    -none-    numeric
```

Finding Optimal Lambda

Again, we used cross validation to find the optimal lambda value that minimizes the mean squared error with the Lasso method.

```
## [1] 0.001123014
```

The optimal λ we found is 0.001123, so we are going to tune our lasso model using this λ value.

In the graph below we plotted $log(\lambda)$ against the mean squared error.

Find Coefficients of Best Model

Next, we view the estimates of the coefficients for the lasso model.

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept)  1.180607e+01
## ExterQual   -7.503821e-02
## LotArea      2.336907e-06
## GarageCars   1.729282e-01
## BsmtFinSF1   1.329442e-04
## HalfBath     7.821342e-02
## FullBath     1.756948e-01
## KitchenQual -6.425955e-02
## OpenPorchSF  8.873340e-05
```

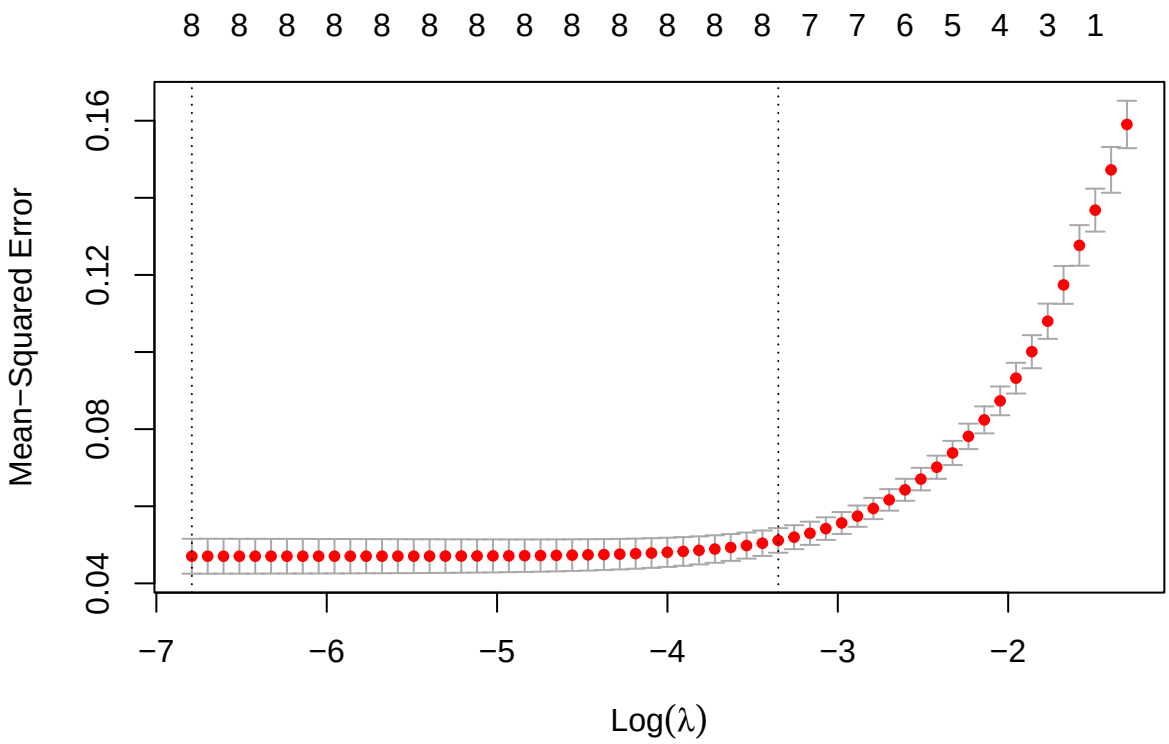

Figure 3: MSE and log(lambda)(Lasso model)

In our case, no coefficient is shrunk all the way to zero, but the coefficients of LotArea, BsmtFinSF1, and OpenPorchSF are very close to zero, which means that they have close to no influence in our lasso model.

So that we could explore lasso regression further, and view what happens as lambda increases, we created the following graph.

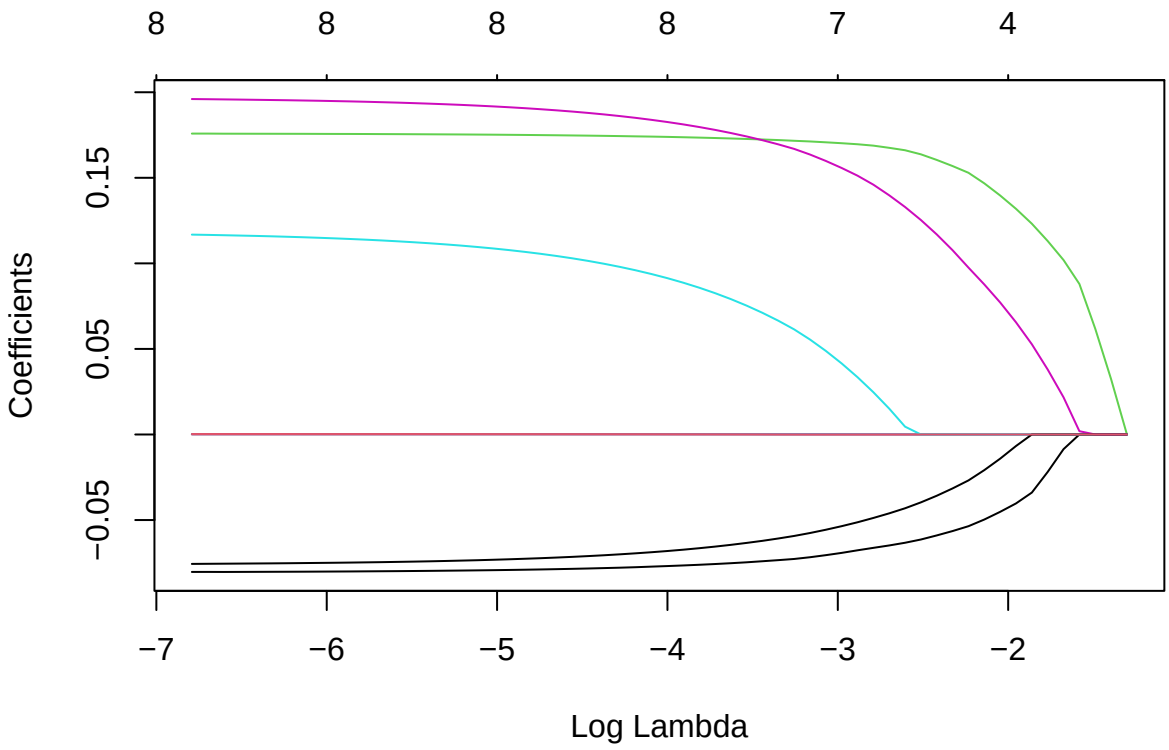

Figure 4: Effect on coefficients as Lambda Increases(Lasso model)

Finally, we wanted to see what the r^2 value is for our lasso regression model.

```
## [1] 0.6972305
```

This value of r^2 means that the best model possible, found as a result of lasso regression, explains 69.72% of the variation in the response.

Graph MLR, RR, LASSO models in a singel graph

The scatter plot below depicts the predicted vs. actual values for the statistical model we obtained in step3(labeled MLR), the ridge(labeled Ridge) and lasso(labeled Lasso) regression models we just obtained:

The MLR model(green) shows a wider spread of predicted values, indicating higher variance in predictions.

The Ridge model(red) has a scatter that clusters closer to the identity line(y=x), suggesting less variance in predictions. This is due to the trade-off between variance and bias that the ridge regression intends to optimize.

The Lasso model (blue) is similar to Ridge but with some points deviating less from the identity line, possibly due to increased shrinkage of the coefficients of some of the variables.

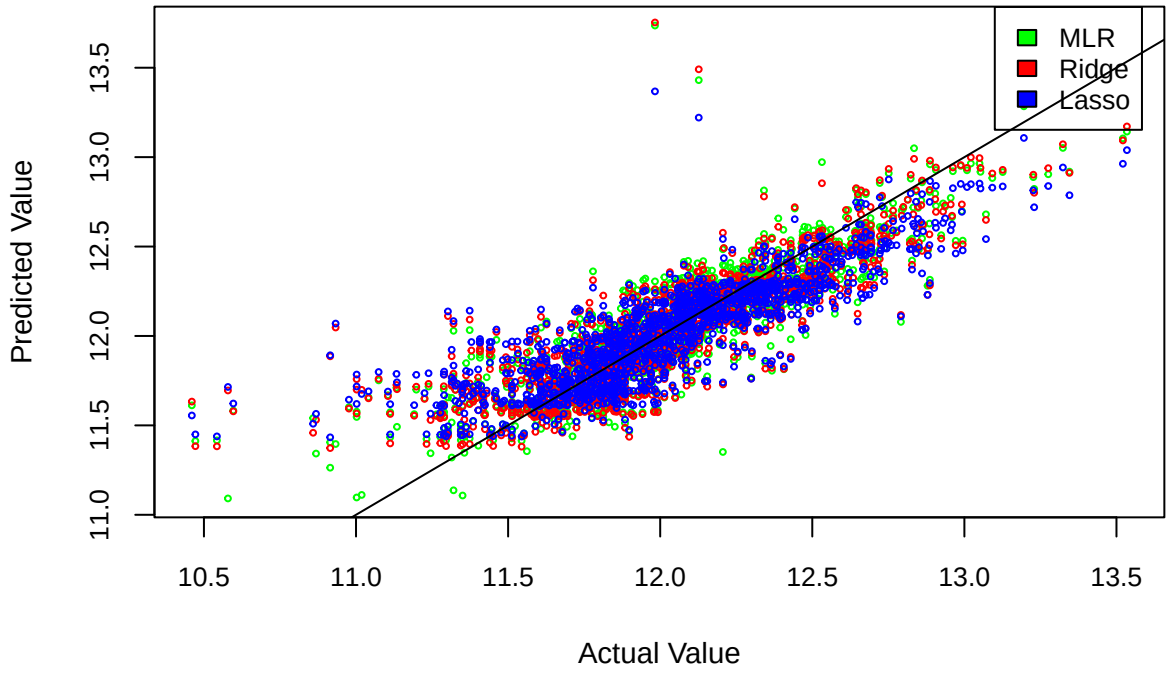

Figure 5: scatter plot of the predicted vs. actual values for MLR, Rridge, and Lasso Models

Conclusion

In summary, the MLR, Ridge, and Lasso models have R-squared values of 0.828, 0.7132, and 0.6972, respectively, suggesting that MLR achieves the best fit. However, by analyzing the predicted vs. response graph, Ridge and Lasso have lower variances in predictions and might generalize better to unseen data due to their reduced complexity.

Some further queries we could investigate would be why the models seem to overestimate/underestimate when the response values are at the lower/higher extremes(as can be seen in the graph above). Also, it came to our surprise that the lasso regression model did not shrink any of our variables to zero, which could also be further investigated.

Innovation: Weighted least square

The analysis technique we have chosen to analyze is the method of weighted least squares. In the previous steps of our project, we explored residual plots and found that our data had heteroscedasticity, or unequal variance. In order to fix that, we took the log of our response variable. Another way in which this issue can be addressed is through weighted linear regression, or the method of weighted least squares. This method places weights on observations so that the ones with smaller error variance are more influential.

Investigate Heteroscedasticity

First, we create our linear model in which we set SalePrice as the response. In the following graph, you can see that there is some fanning of our data points, showing that there is heteroscedasticity.

```
## integer(0)
```

Summary of Linear Model

The following shows the summary of our linear model, before adding weights.

```
##
## Call:
## lm(formula = SalePrice ~ ., data = h2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -404433  -22180   -2143   18266  343668
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.989e+05  1.021e+04  19.468  < 2e-16 ***
## LotArea           9.452e-01  1.127e-01   8.389  < 2e-16 ***
## GarageCars        2.771e+04  2.306e+03  12.016  < 2e-16 ***
## BsmtFinSF1        3.128e+01  2.646e+00  11.824  < 2e-16 ***
## FullBath          2.721e+04  2.698e+03  10.086  < 2e-16 ***
## ExterQualFa      -1.026e+05  1.425e+04  -7.200 9.69e-13 ***
## ExterQualGd      -5.507e+04  7.190e+03  -7.660 3.40e-14 ***
## ExterQualTA      -8.018e+04  7.768e+03 -10.321  < 2e-16 ***
## HouseStyle1.5Unf -6.933e+03  1.160e+04  -0.598 0.550206
```

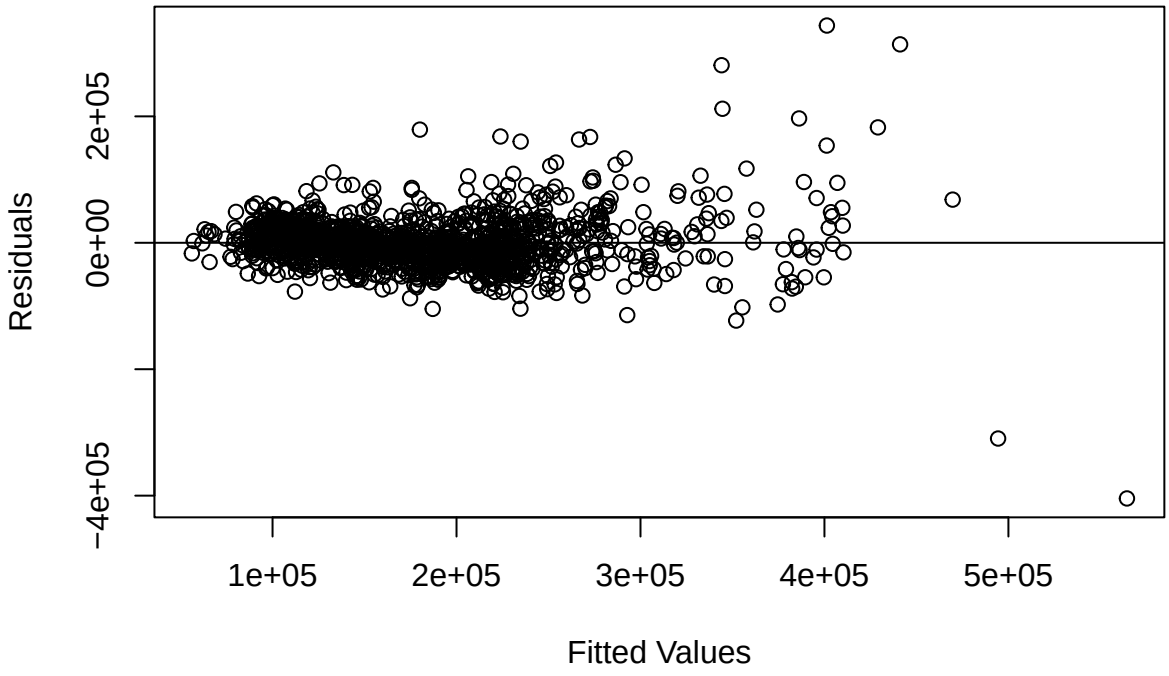

Figure 6: Residual vs. fitted graph for linear model without log transformation

```
## HouseStyle1Story -7.205e+03  3.965e+03  -1.817 0.069399 .
## HouseStyle2.5Fin  4.193e+04  1.508e+04   2.780 0.005502 **
## HouseStyle2.5Unf  2.662e+03  1.302e+04   0.205 0.837984
## HouseStyle2Story -7.092e+03  4.367e+03  -1.624 0.104593
## HouseStyleSFoyer -2.380e+04  7.852e+03  -3.031 0.002478 **
## HouseStyleSLvl   -3.519e+03  6.285e+03  -0.560 0.575611
## OpenPorchSF       2.964e+01  1.768e+01   1.676 0.093974 .
## HalfBath          1.722e+04  2.964e+03   5.809 7.74e-09 ***
## KitchenQualFa    -7.112e+04  9.307e+03  -7.642 3.90e-14 ***
## KitchenQualGd    -5.404e+04  5.415e+03  -9.980  < 2e-16 ***
## KitchenQualTA    -7.122e+04  5.950e+03 -11.970  < 2e-16 ***
## GarageFinishNA    6.258e+03  6.886e+03   0.909 0.363641
## GarageFinishRFn  -5.721e+03  3.074e+03  -1.861 0.062915 .
## GarageFinishUnf  -1.225e+04  3.433e+03  -3.570 0.000369 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41080 on 1437 degrees of freedom
## Multiple R-squared:  0.7366, Adjusted R-squared:  0.7326
## F-statistic: 182.7 on 22 and 1437 DF,  p-value: < 2.2e-16
```

As you can see, the residual standard error is 41080, and the r^2 value is 73.66%.

Add Weights

To address the potential heteroscedasticity, we define weights(wt) inversely proportional to the squared fitted values of the initial model(lmod). This assigns higher weights to observations with smaller residuals, reducing the influence of potentially less reliable data points.

```
#define weights to use
wt <- 1 / lm(abs(lmod$residuals) ~ lmod$fitted.values)$fitted.values^2

#perform weighted least squares regression
wls_model <- lm(SalePrice ~ ., data = h2, weights=wt)
```

View the Summary of the Model, Now with Weights The following shows the summary of our model, with the weights.

```
##
## Call:
## lm(formula = SalePrice ~ ., data = h2, weights = wt)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -7.0727 -0.7971 -0.0661  0.7172  6.7485
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.255e+05  1.263e+04  17.849  < 2e-16 ***
## LotArea           1.911e+00  1.534e-01  12.459  < 2e-16 ***
## GarageCars        1.318e+04  1.719e+03   7.668 3.20e-14 ***
## BsmtFinSF1        3.143e+01  2.347e+00  13.392  < 2e-16 ***
```

```
## FullBath          2.363e+04  1.958e+03  12.069  < 2e-16 ***
## ExterQualFa      -1.190e+05  1.283e+04  -9.273  < 2e-16 ***
## ExterQualGd      -7.269e+04  1.217e+04  -5.972 2.95e-09 ***
## ExterQualTA      -9.836e+04  1.230e+04  -7.998 2.59e-15 ***
## HouseStyle1.5Unf -1.726e+03  5.048e+03  -0.342 0.732457
## HouseStyle1Story -7.743e+03  2.141e+03  -3.617 0.000308 ***
## HouseStyle2.5Fin  3.274e+04  1.388e+04   2.358 0.018499 *
## HouseStyle2.5Unf  1.039e+04  7.598e+03   1.368 0.171541
## HouseStyle2Story -2.351e+03  2.765e+03  -0.851 0.395182
## HouseStyleSFoyer -1.021e+04  3.767e+03  -2.712 0.006770 **
## HouseStyleSLvl    2.319e+03  4.195e+03   0.553 0.580482
## OpenPorchSF       3.130e+01  1.332e+01   2.350 0.018905 *
## HalfBath          1.488e+04  2.095e+03   7.101 1.93e-12 ***
## KitchenQualFa    -6.228e+04  8.080e+03  -7.708 2.38e-14 ***
## KitchenQualGd    -4.149e+04  7.438e+03  -5.578 2.90e-08 ***
## KitchenQualTA    -5.911e+04  7.515e+03  -7.866 7.17e-15 ***
## GarageFinishNA   -1.614e+04  4.065e+03  -3.970 7.55e-05 ***
## GarageFinishRFn  -3.754e+03  2.924e+03  -1.283 0.199528
## GarageFinishUnf  -1.437e+04  2.802e+03  -5.129 3.32e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.363 on 1437 degrees of freedom
## Multiple R-squared:  0.7501, Adjusted R-squared:  0.7463
## F-statistic: 196.1 on 22 and 1437 DF,  p-value: < 2.2e-16
```

As you can see, with the weighted model, the residual standard error is 1.363, and the r^2 value is 75.01%.

Check assumptions of the weghted least sqaure model Since this method is used to address heteroscedasticity, we do not assume constant variance of the error terms in this model. However, we still need to check the other two assumptions in the OLS model, which are linearity and zero expectation of the error terms. We will do this by checking the resdual plots of the weghted model:

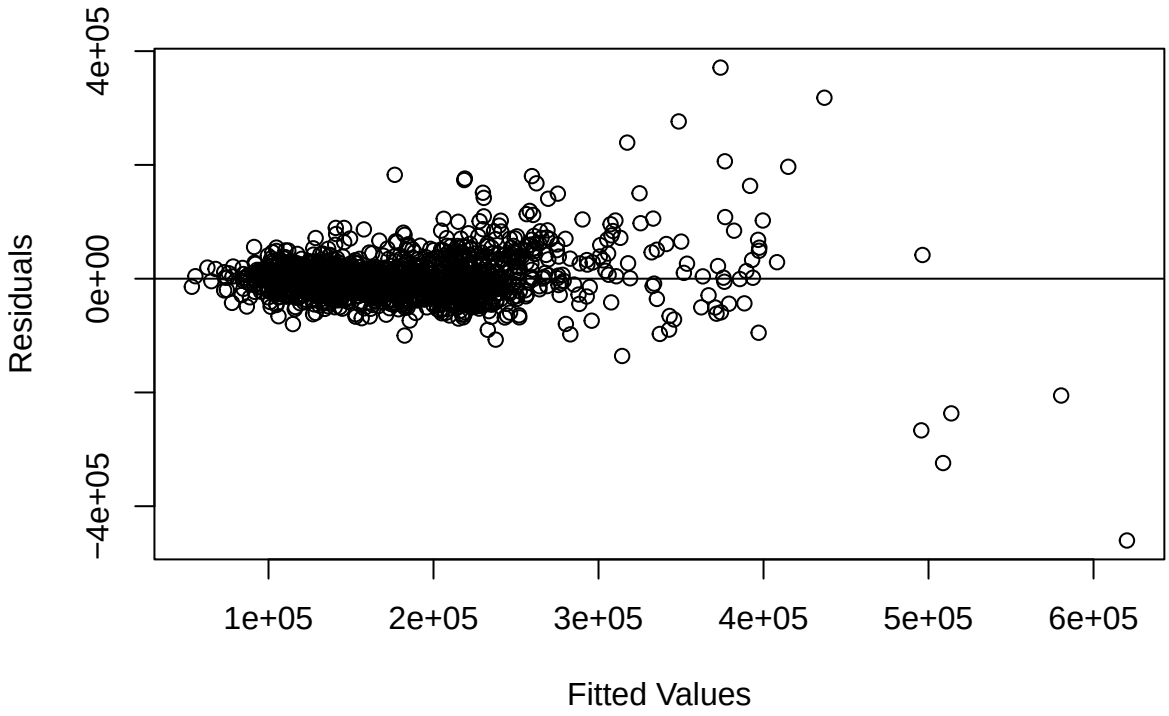

Figure 7: Residual vs. fitted graph for linear model without log transformation

As we can see from the residual vs. fitted plot there is no notable patterns suggesting that the linearity assumption is violated. Also vast majority of the residuals spread evenly above and below the zero line so we can assume that the expectation of the error term is zero in this case.

Compare With and Without Weights

Comparing the summaries of lmod and wls_model, we observed changes in coefficient estimates, standard errors and r^2 values. Especially, the residual standard error shows a drastic change from the model without weights. This means that the values that are predicted with the weighted model are much more accurate and aligned with the actual observations. Also, because the r^2 value increased, we know that the weighted model is able to explain more of the variance in SalePrice.