**House Price Data Challenge**

**The Cube:** Sicheng Chu, Yiwen Wang, Luke Zheng

**Introduction**

There is a lot of big money to be made in real estate. Whether it be through flipping homes, renting out buildings, or simply buying low and selling high, a model that accurately predicts the value of homes is invaluable. Our goal for this project is to create a robust and relatively interpretable model for predicting the possible sale price of a home given various characteristics including square footage, location, build style, etc. We aimed to select as few predictors as we feel are necessary to increase interpretability and prevent overfitting.

**Methods**

Since we are picking from a dataset that has 79 predictors, we needed to narrow down the variables we wanted to use. This also prevents the curse of dimensionality which arises when too many variables are used in a linear model, which results in random variation in the response variable being mistaken for effects caused by predictors.

We conducted multiple initial simple linear regressions with predictors we were curious about based on our intuition to get a general sense of what variables may be good predictors. We also disregarded variables that were too uninteresting and complicated such as the variables named “Condition1” and “Condition2”. Other categorical variables that had very few occurrences of various levels within the variable were disregarded. One particular case of this was with the variable “Utilities” where the majority of observations fell into one category(“AllPub”)and only one observation fell into the other one(“NoSeWa”). Therefore, this variable was not included in the model.

Collinearity was a concern since some predictors were definitely not independent of each other. We attempted to avoid this issue by using boxplots and chi-square tests on categorical variables, and doing simple linear regression models and correlation graphs on quantitative variables to select the most important and independent features.

The second part of our variable selection process involved 5-fold cross-validation to prevent our model from overfitting any of our training data. To do so, we wrote R code that withheld a fifth of the training dataset to be used as a test dataset, while the remaining is used to train the data to our linear model. When the model is trained to the training data, we try to predict the values of the test dataset, and from that we can compute a mean squared error. Our two most interesting cross-validations involved comparing a linear model predicting the sale price vs a model predicting the log sale price. Since the distribution of the original response variable sale price is highly skewed, we used the log transformation to make it better fit the assumptions underlying regression. The mean squared errors from the model predicting the absolute sale price were slightly higher than the mean squared errors from the model predicting the log sale price, which means the model predicting for the log sale price is overall better at predicting accurately. Hence, we decided to train our model for predicting the log of the sale price.

As a result, we ended up with 15 predictors we wanted in our model. Among these variables, we customized some categorical variables that had levels corresponding to the grading scale of some qualities, changing these to be no (if the result is below average) and yes (otherwise). And we standardized quantitative variables with large units. The summary and the list of the predictors will be presented in the “Results” section.

**Results**

We ran the entirety of our training data through a multiple linear regression in R to predict the log sale price of homes, which resulted in the linear model summary shown on the following page. When we submitted our predictions on Kaggle, we got a root mean squared error of 0.14709 and the model had an adjusted R2 = 0.879. Linear model output is in the reference page. The residuals plotted against the fitted values appear to be randomly distributed without any noticeable heteroskedasticity. The residuals also appear to be normally distributed. Plots are on the next page.

**Conclusion**

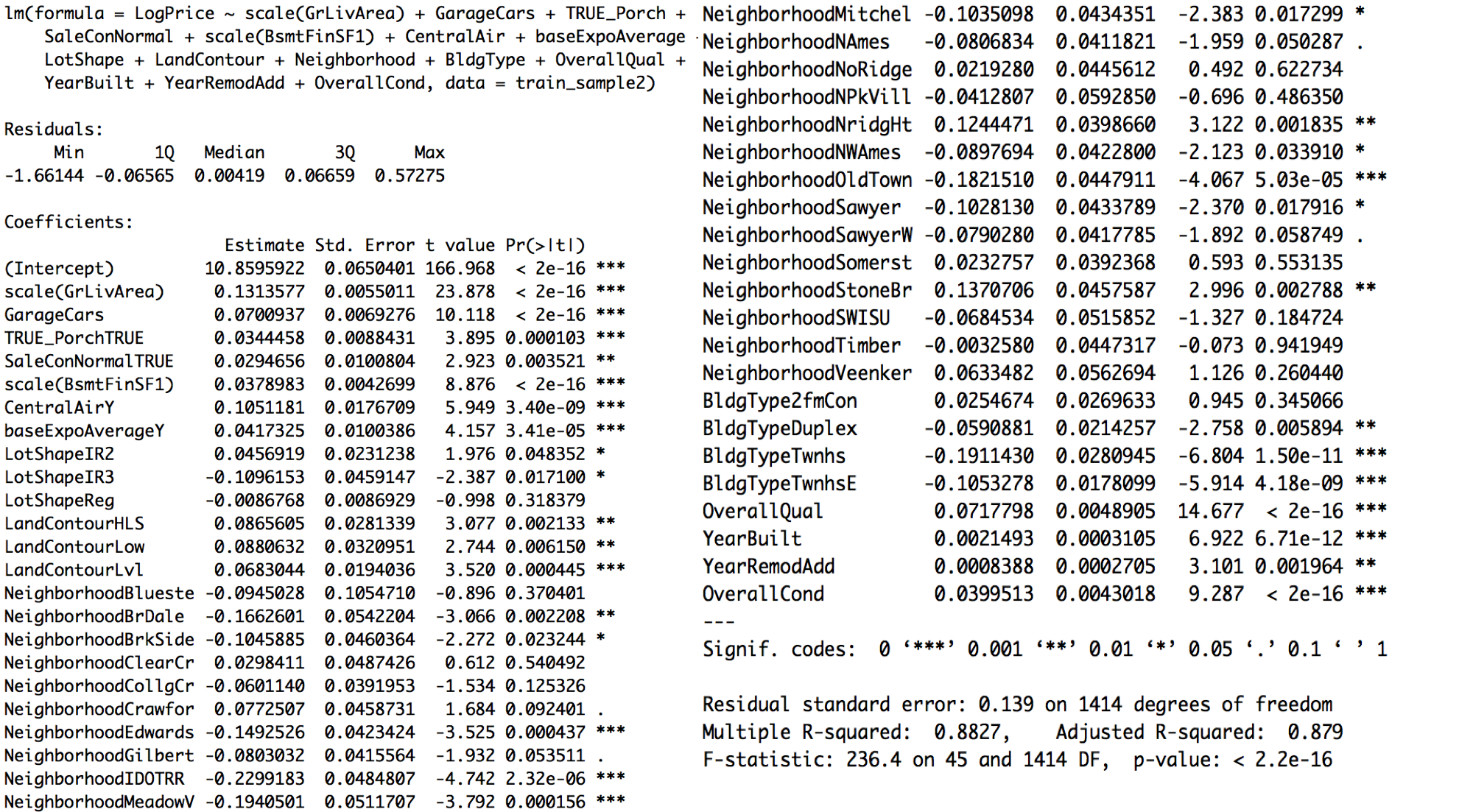
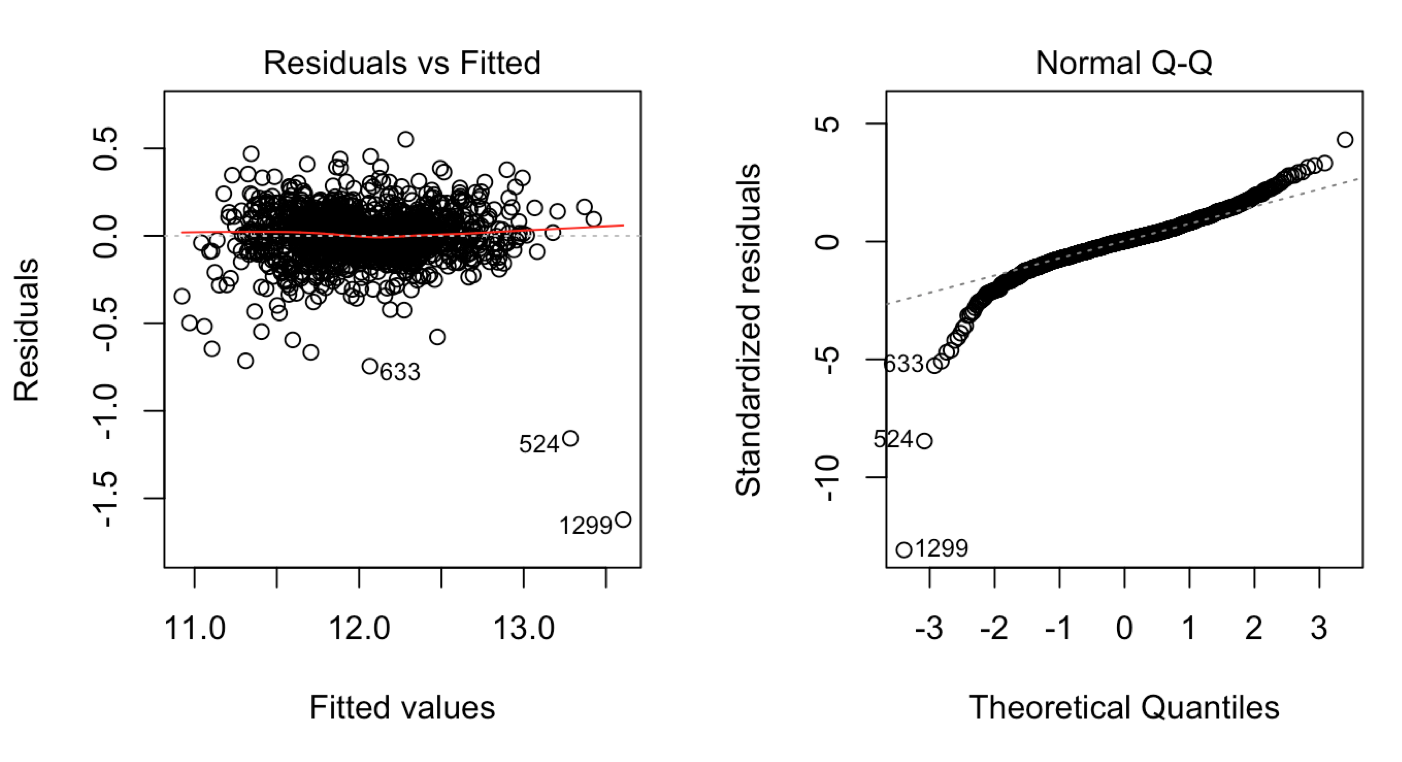
One of the most noticeable results from the linear model is the effect certain neighborhoods have on the price. Since we are predicting the log sale price, the beta coefficients correspond to an times increase in price. Neighborhoods such

as Edwards, Meadow Village, Old Town, and Briardale have coefficients around -0.15 which correspond to an approximate 14% decrease in sale price (). Stone Brook has a coefficient around 0.13 which corresponds to an approximate 13% increase in price. All of these coefficients have low p-values. Hence houses in Stone Brook are going to be noticeably more expensive than those of the former neighborhoods.

**Possible Future Directions**

During our initial simple linear regressions, we noticed that variables such as “BsmtUnfSF” and “BsmtFinSF1” have many values at 0 since there are many houses that simply do not have basements, so naturally their square footages would be 0. However even though these houses do not have basements, their prices could still vary a great deal based on their other features. Hence, an interesting analysis we could do would be conditional linear regressions on houses with basements and houses without basements.

Similarly, other variables such as “GarageArea” produced NA’s when the house has no garage, hence it conditional linear model on the presence of garages would also be an interesting future direction.

Furthermore, if given more time, we could also include quadratic or interaction terms into our model and run them through the cross-validation to test their robustness and accuracy.

**After Game:**

What can I get for the audience from the data?

Visualization of data:supported by the data. Explain the data and table. Put the most relevant visualization.