STAT 420 Su 2019 Final Project

Shane Taylor shanemt2@illinois.edu (mailto:shanemt2@illinois.edu); Zhouning Ma zm11@illinois.edu (mailto:zm11@illinois.edu); Kevin Mackie kevindm2@illinois.edu (mailto:kevindm2@illinois.edu)

8/3/2019

Group members

- Shane Taylor shanemt2@illinois.edu (mailto:shanemt2@illinois.edu)
- Zhouning Ma zm11@illinois.edu (mailto:zm11@illinois.edu)
- Kevin Mackie kevindm2@illinois.edu (mailto:kevindm2@illinois.edu)

Introduction

This project is a linear regression analysis of housing prices, based on a publicly available data set on housing prices in the city of Ames, Iowa, that was curated by Dean De Cock of Truman state university.

- https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview (https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview)
- http://jse.amstat.org/v19n3/decock.pdf (http://jse.amstat.org/v19n3/decock.pdf)

For this project, our goal will be to attempt to find, using a variety of statistical modelling techniques in R, a “good” (below) linear regression model that both performs well and is easily interpretable for explanatory purposes.

A “**good model**” should:

- Have a Mean Absolute Percentage Error less than 15%, and preferably 10% or lower
- Adhere to LINE assumptions for linear regression i.e. has constant variance and demonstrates normality
- Has no destabilizing collinearity among the predictors i.e. no VIF >= 5

Data set

The data set consists of 1460 observations with one numerical response (house price) and 80 predictors. The predictors are a relatively even mix of numerical and categorical variables.

The predictors are summarized in the following table:

Variable name	Description	Factor Levels
SalePrice	the property’s sale price in dollars. This is the target variable that you’re trying to predict.	
mssubclass	the building class	
MSZoning	The general zoning classification	C (all), FV, RH, RL, RM
LotFrontage	Linear feet of street connected to property	
LotArea	Lot size in square feet	

Variable name	Description	Factor Levels
Street	Type of road access	Grvl, Pave
Alley	Type of alley access	Grvl, Pave
LotShape	General shape of property	IR1, IR2, IR3, Reg
LandContour	Flatness of the property	Bnk, HLS, Low, Lvl
Utilities	Type of utilities available	AllPub, NoSeWa
LotConfig	Lot configuration	Corner, CulDSac, FR2, FR3, Inside
LandSlope	Slope of property	Gtl, Mod, Sev
Neighborhood	Physical locations within Ames city limits	Blmngtn, Blueste, …
Condition1	Proximity to main road or railroad	Artery, Feedr, Norm, PosA, …
Condition2	Proximity to main road or railroad (if a second is present)	Artery, Feedr, …
BldgType	Type of dwelling	1Fam, 2fmCon, Duplex, Twnhs, TwnhsE
HouseStyle	Style of dwelling	1.5Fin, 1.5Unf, 1Story, 2.5Fin, 2.5Unf, 2Story, SFoyer, SLvl
OverallQual	Overall material and finish quality	
OverallCond	Overall condition rating	
YearBuilt	Original construction date	
YearRemodAdd	Remodel date	
RoofStyle	Type of roof	Flat, Gable, Gambrel, Hip, Mansard, Shed
RoofMatl	Roof material	ClyTile, CompShg, Membran, Metal, Roll, Tar&Grv, WdShake, WdShngl
Exterior1st	Exterior covering on house	AsbShng, AsphShn, BrkComm, BrkFace, CBlock, …
Exterior2nd	Exterior covering on house (if more than one material)	AsbShng, AsphShn, …
MasVnrType	Masonry veneer type	BrkCmn, BrkFace, None, Stone
MasVnrArea	Masonry veneer area in square feet	
ExterQual	Exterior material quality	Ex, Fa, Gd, TA
ExterCond	Present condition of the material on the exterior	Ex, Fa, Gd, Po, TA
Foundation	Type of foundation	BrkTil, CBlock, PConc, Slab, Stone, Wood
BsmtQual	Height of the basement	Ex, Fa, Gd, TA
BsmtCond	General condition of the basement	Fa, Gd, Po, TA
BsmtExposure	Walkout or garden level basement walls	Av, Gd, Mn, No

Variable name	Description	Factor Levels
BsmtFinType1	Quality of basement finished area	ALQ, BLQ, GLQ, LwQ, Rec, Unf
BsmtFinSF1	Type 1 finished square feet	
BsmtFinType2	Quality of second finished area (if present)	ALQ, BLQ, GLQ, LwQ, Rec, Unf
BsmtFinSF2	Type 2 finished square feet	
BsmtUnfSF	Unfinished square feet of basement area	
TotalBsmtSF	Total square feet of basement area	
Heating	Type of heating	Floor, GasA, GasW, Grav, OthW, Wall
HeatingQC	Heating quality and condition	Ex, Fa, Gd, Po, TA
CentralAir	Central air conditioning	N, Y
Electrical	Electrical system	FuseA, FuseF, FuseP, Mix, SBrkr
1stFlrSF	First Floor square feet	
2ndFlrSF	Second floor square feet	
LowQualFinSF	Low quality finished square feet (all floors)	
GrLivArea	Above grade (ground) living area square feet	
BsmtFullBath	Basement full bathrooms	
BsmtHalfBath	Basement half bathrooms	
FullBath	Full bathrooms above grade	
HalfBath	Half baths above grade	
Bedroom	Number of bedrooms above basement level	
Kitchen	Number of kitchens	
KitchenQual	Kitchen quality	Ex, Fa, Gd, TA
TotRmsAbvGrd	Total rooms above grade (does not include bathrooms)	
Functional	Home functionality rating	Maj1, Maj2, Min1, Min2, Mod, Sev, Typ
Fireplaces	Number of fireplaces	
FireplaceQu	Fireplace quality	Ex, Fa, Gd, Po, TA
GarageType	Garage location	2Types, Attchd, Basment, BuiltIn, CarPort, Detchd
GarageYrBlt	Year garage was built	
GarageFinish	Interior finish of the garage	Fin, RFn, Unf

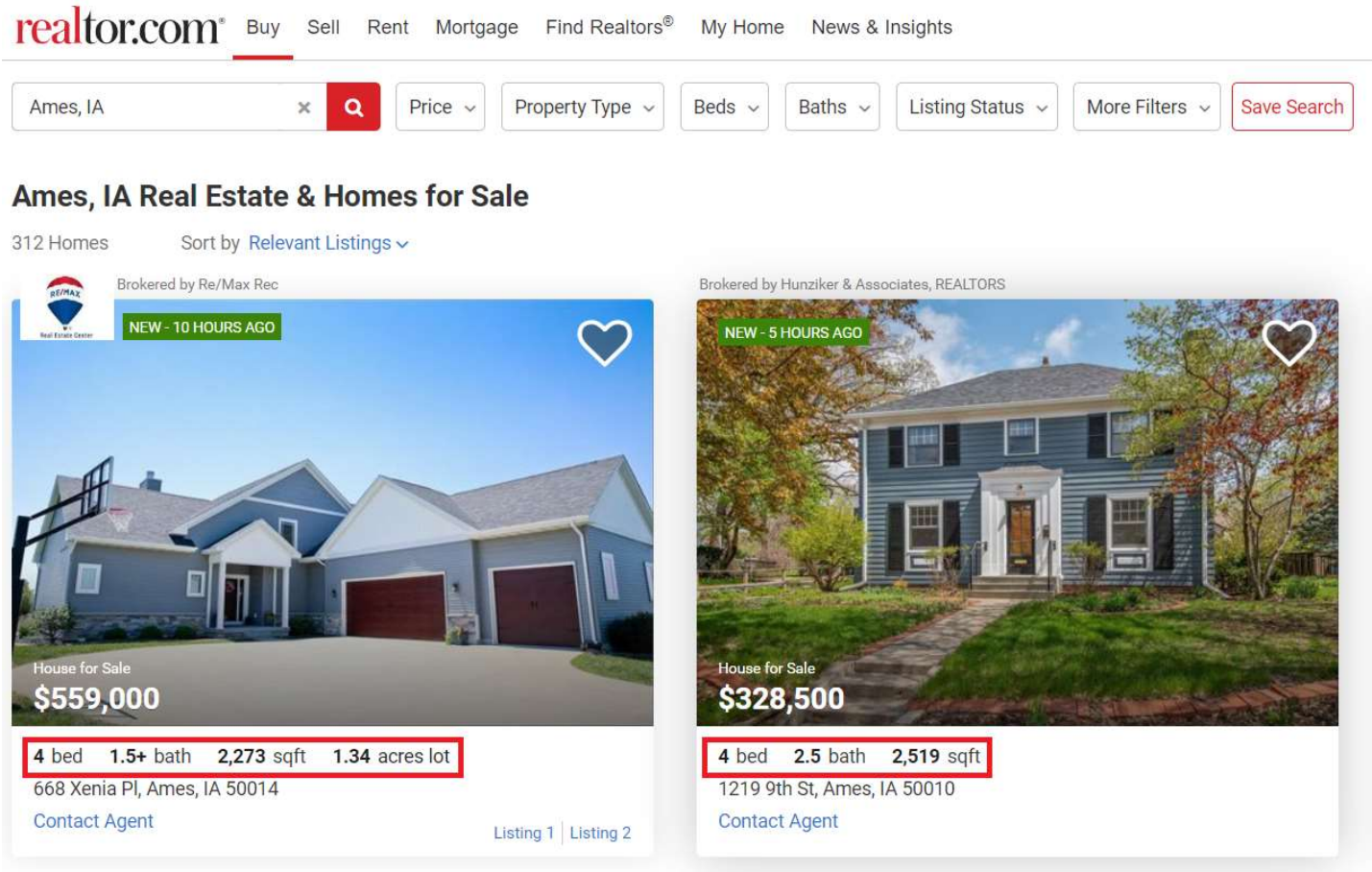
Variable name	Description	Factor Levels
GarageCars	Size of garage in car capacity	
GarageArea	Size of garage in square feet	
GarageQual	Garage quality	Ex, Fa, Gd, Po, TA
GarageCond	Garage condition	Ex, Fa, Gd, Po, TA
PavedDrive	Paved driveway	N, P, Y
WoodDeckSF	Wood deck area in square feet	
OpenPorchSF	Open porch area in square feet	
EnclosedPorch	Enclosed porch area in square feet	
3SsnPorch	Three season porch area in square feet	
ScreenPorch	Screen porch area in square feet	
PoolArea	Pool area in square feet	
PoolQC	Pool quality	Ex, Fa, Gd
Fence	Fence quality	GdPrv, GdWo, MnPrv, MnWw
MiscFeature	Miscellaneous feature not covered in other categories	Gar2, Othr, Shed, TenC
MiscVal	$Value of miscellaneous feature	
MoSold	Month Sold	
YrSold	Year Sold	
SaleType	Type of sale	COD, Con, ConLD, ConLI, ConLw, CWD, New, Oth, WD
SaleCondition	Condition of sale	Abnorml, AdjLand, Alloca, Family, Normal, Partial

Methods

Approach to Problem (Utilizing Real Estate Websites)

$Hypothesis:$ Using only the characteristics most prominently displayed across typical real estate websites, we can build a "good" (as described above) linear model for predicting home prices.

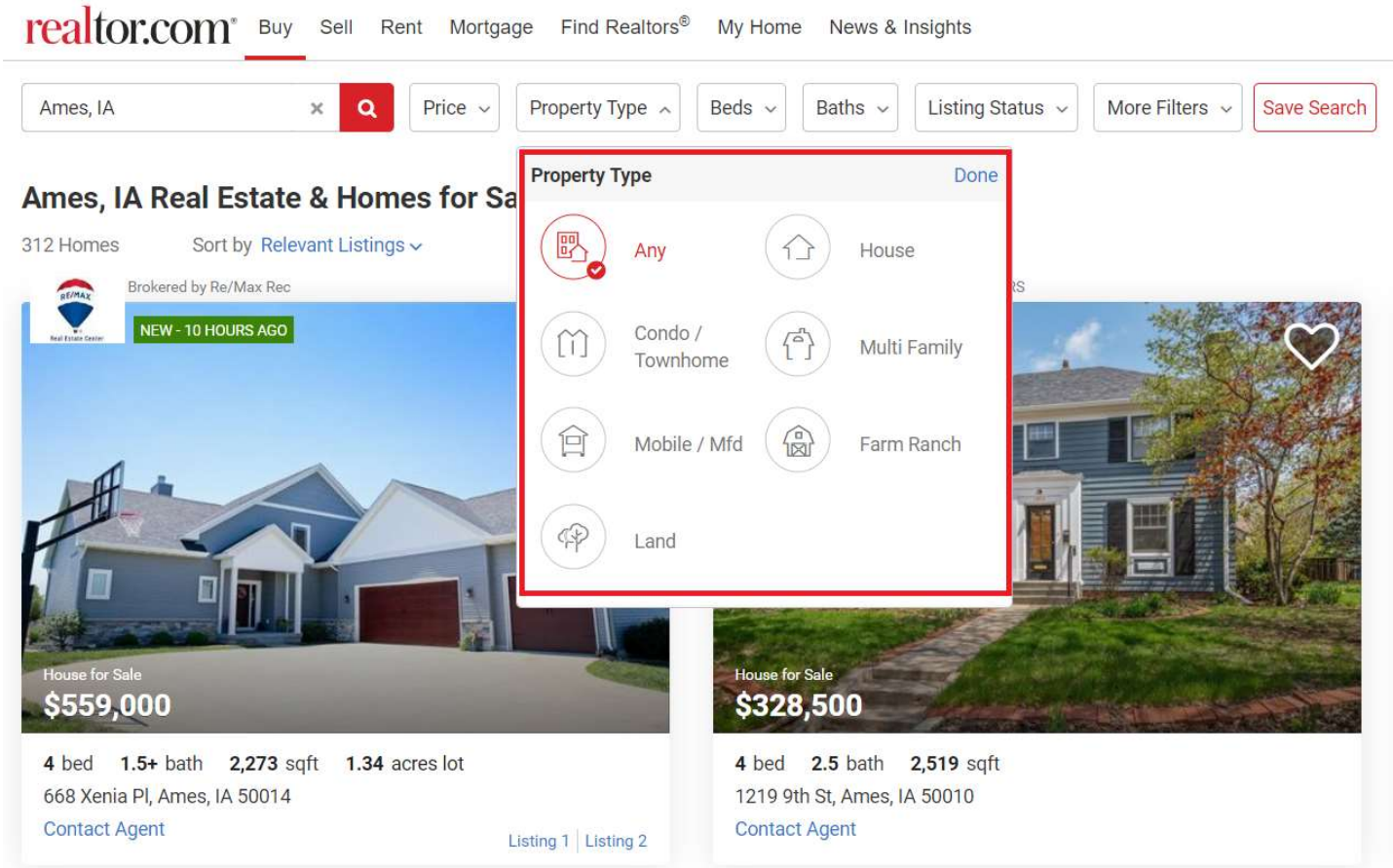
Looking at the default view of some of the listings on realtor.com, we can see that the following information is displayed.



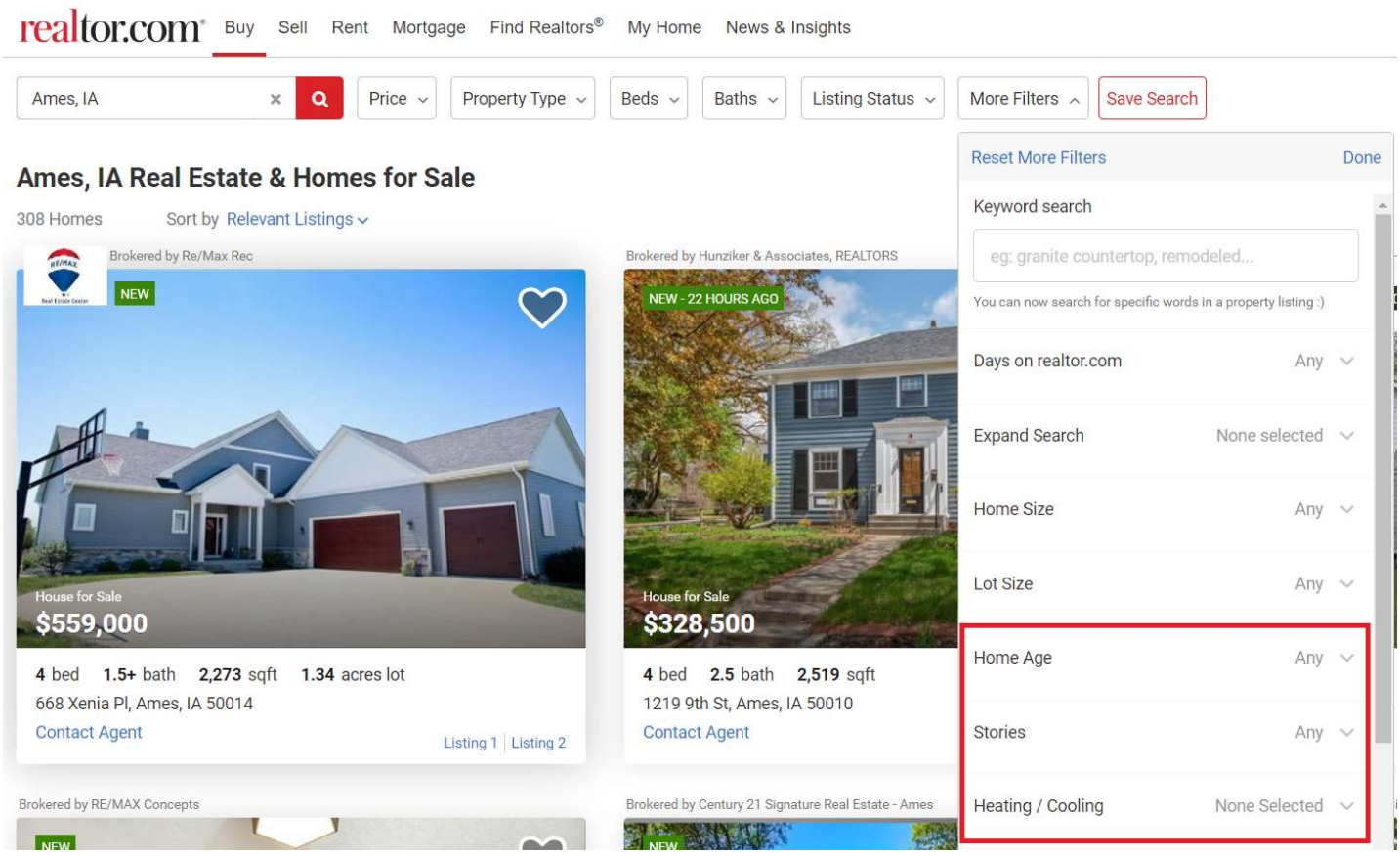
- # of bed rooms
- # of bath rooms
- Total square footage
- Lot square footage (some listings only)

Other realtor websites such as zillow.com, redfin.com, etc. also show similar data for each listing. The fact that all of these different websites display the same default data tells us that it is likely that these are the most important aspects of a house for a potential buyer. Under our hypothesis, we assume these are highly correlated with the sale price.

Additionally, the various realtor websites allow one specify the dwelling type, such as condo, house, aparment, etc. This suggests these will be important characteristics as well.



The top options under the website filters include the age of home, the number of stories, and whether it has heating or cooling.

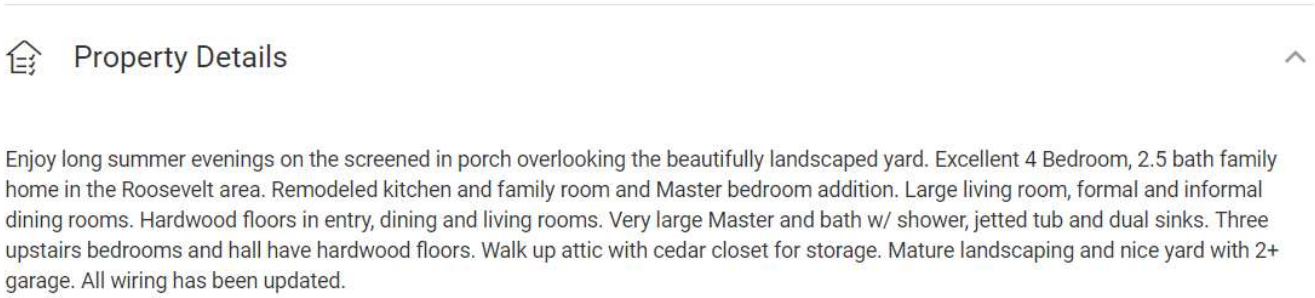


Taking all of this into account, we will build a model using the information most prominently featured on realtor websites. The initial numeric variables that we will use in this approach are:

- square footage of the house
- lot size
- age of the home
- number of bathrooms and bedrooms.

We consider a small subset of factor variables that seem the most likely to determine price, these include type of building (`BldgType`), the style of the home (`HouseStyle`), and the inclusion of heating and central AC.

Also, note that two of the variables in the data are `OverallQual` and `OverallCond` . It's not too much of a stretch to assume that the overall quality of the construction of the house, and the condition of the house would also factor into the asking price for the home. But the way this information is displayed on the website is not with a number, but typically with a paragraph or two, such as:



We will be using the presence of descriptive paragraphs such as these to support my ability to include these two variables under my hypothesis. Essentially we will be assuming that the score given by `OverallCond` and `OverallQual` is more or less an accurate heuristic for converting these sorts of descriptions into a number.

Library

```
library(tidyverse)
library(lmtest)
library(faraway)
library(MASS)
library(corpcor)
library(corrplot)
library(ggcorrplot)

library(GGally)
library(leaps)
library(knitr)
library(kableExtra)
library(caret)
library(outliers)
library(ggplot2)
library(ggthemes)
```

Public Functions

```
diagnostics = function(model, pcol = "grey", lcol = "dodgerblue"){
  par(mfrow = c(1,2))
  plot(fitted(model), resid(model), col = pcol, pch = 20,
       xlab = "Fitted", ylab = "Residuals", main = paste("Fitted Vs Residuals"))
  abline(h = 0, col = lcol, lwd = 2)
  qqnorm(resid(model), col = pcol)
  qqline(resid(model), lty = 2, lwd = 2, col = lcol)
}

mape = function(actual, pred){
  mean(abs((actual - pred) / actual)) * 100
}

get_loocv_rmse = function(model) {
  sqrt(mean((resid(model) / (1 - hatvalues(model))) ^ 2))
}

get_adj_r2 = function(model) {
  summary(model)$adj.r.squared
}

kfold_cv = function(data, model, k, log_response = FALSE) {
  set.seed(600)
  # Get indices for k folds (the data to be held out)
  holdouts = split(sample(1:nrow(data)), 1:k)

  total_mape = 0

  # Now for each holdout fit a model to the data not held out, and test against held out data
  for (holdout in holdouts) {
    train = data[-c(holdout),]
    test = data[c(holdout),]
    m = lm(formula(model), data = train)
    pred = predict(m, newdata = test)
    if (log_response) {
      total_mape = total_mape + mape(test$SalePrice, exp(pred))
    } else {
      total_mape = total_mape + mape(test$SalePrice, pred)
    }
  }
  total_mape / k
}
```

Loading the data

We load the “train.csv” file and observe the available variables.

```
house_prices_full = read.csv("train.csv")
#str(house_prices_full)
```

Note that there is also a “test.csv” file, but it does not contain the `SalePrice` response variable. We split the data from “train.csv” randomly into a 75/25 train/test split as `house_prices` and `house_prices_test`. We will train our models on `house_prices` and evaluate our final model in terms of how well it works to make predictions on `house_prices_test`.

```
set.seed(08032019)
house_prices_idx  = sample(nrow(house_prices_full), size = trunc(0.75 * nrow(house_prices_full)))
house_prices = house_prices_full[house_prices_idx, ]
house_prices_test = house_prices_full[-house_prices_idx, ]
```

Initial variable selection and additive model

Based on our above real-world assumptions and review of realtor websites, we make the following initial variable selection:

- GrLivArea: Above grade (ground) living area square feet
- LotArea: Lot size in square feet
- TotalBsmtSF: Total square feet of basement area
- BsmtUnfSF: Unfinished square feet of basement area
- GarageArea: Size of garage in square feet
- BedroomAbvGr: Number of bedrooms above basement level
- FullBath: Full bathrooms above grade
- HalfBath: Half baths above grade
- BsmtFullBath: Basement full bathrooms
- BsmtHalfBath: Basement half bathrooms
- BldgType: Type of dwelling : factors = { 1Fam, 2fmCon, Duplex, Twnhs, TwnhsE }
- HouseStyle: Style of dwelling : factors = { 1.5Fin, 1.5Unf, 1Story, 2.5Fin, 2.5Unf, 2Story, SFoyer, SLvl }
- OverallQual: Overall material and finish quality
- OverallCond: Overall condition rating
- YearBuilt: Original construction date
- Heating: Type of heating : factors = { Floor, GasA, GasW, Grav, OthW, Wall }
- CentralAir: Central air conditioning : factors = { N, Y }

We first build a large additive model that includes all of the above predictors.

```
house_prices.lm = lm(SalePrice ~ GrLivArea + LotArea + GarageArea + BedroomAbvGr + FullBath + HalfBath +
TotalBsmtSF + OverallQual + OverallCond + YearBuilt + BldgType + HouseStyle + CentralAir + Heating, data
= house_prices)
```

Mean absolute percentage error

We then test this model using a training data set and a k-fold cross validation.

```
(mape1 = kfold_cv(house_prices, house_prices.lm, 5))
```

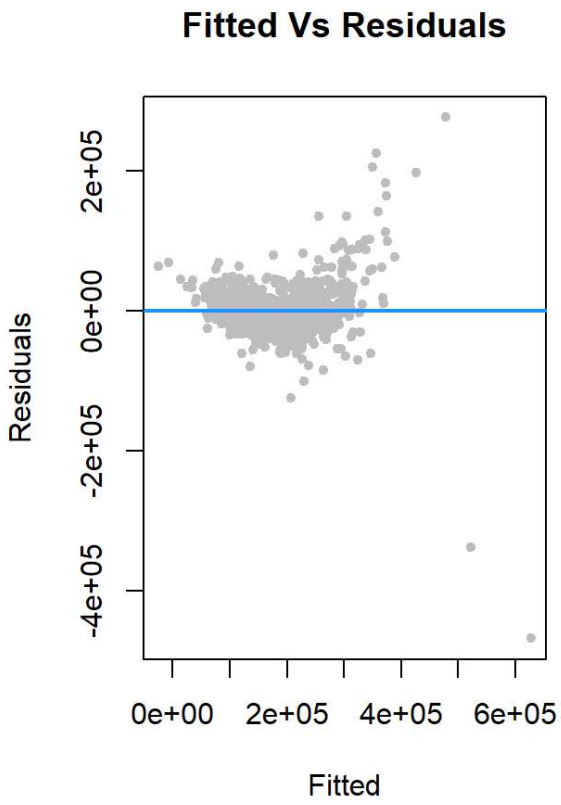
```
## [1] 13.42263
```

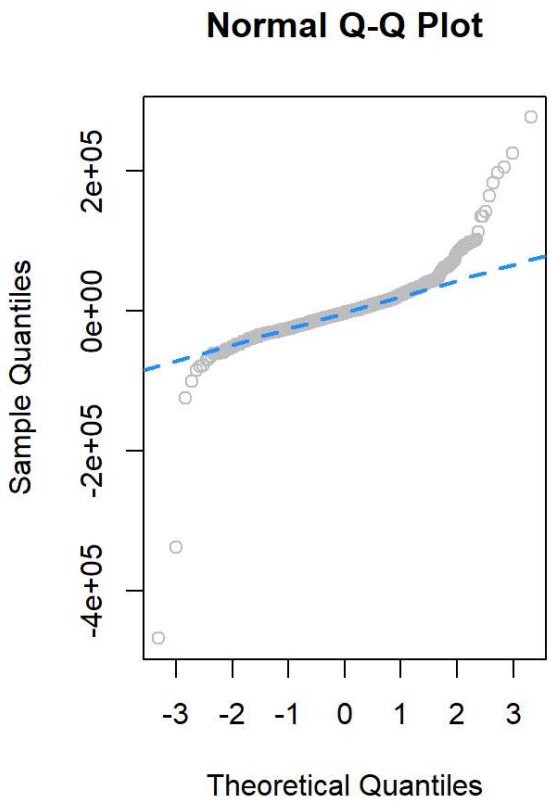
The reported error for this initial model is 13.4226254.

Model diagnostics

Let's look at the Fitted Vs Residuals Plot and the Normal Q-Q Plot.

```
diagnostics(house_prices.lm)
```



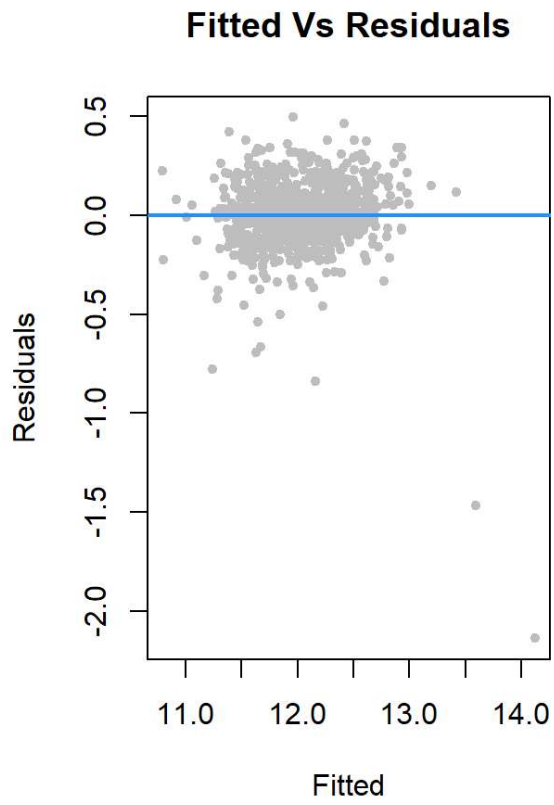


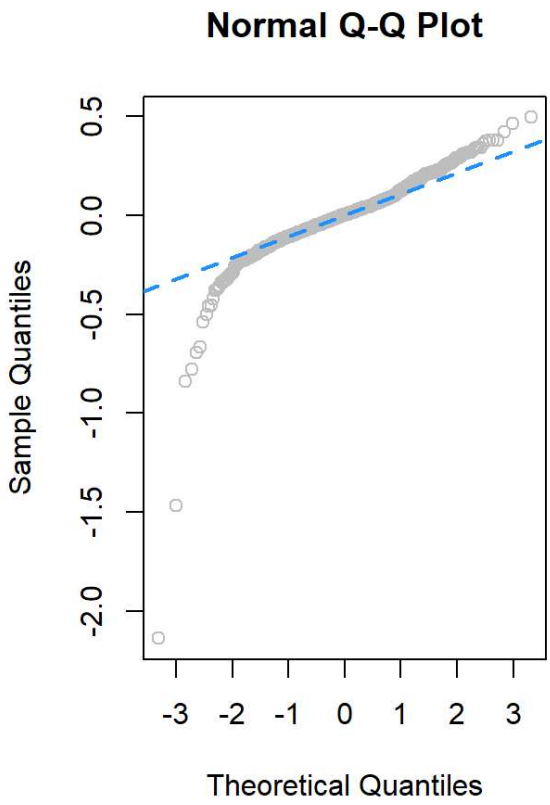
We can see that the constant variance assumption seems to be violated, since the variation increases as you move to the right, The fitted vs residuals plot also exhibits a crescent shape indicating non-linearly. It seems clear that the there is an underlying pattern to the variance, indicating that a transformation might be applicable. Since the variation seems to grow exponentially as it moves to the right, we suspect that a log transformation of the reponse may be effective.

Log transform of response

We fit a model using the log of the response (sale price), and assess the diagnostic plots for the log-response model.

```
house_prices.lm = lm(log(SalePrice) ~ GrLivArea + LotArea + GarageArea + BedroomAbvGr + FullBath + HalfB
ath + TotalBsmtSF + OverallQual + OverallCond + YearBuilt + BldgType + HouseStyle + CentralAir + Heatin
g, data = house_prices)
diagnostics(house_prices.lm)
```





It appears that a log transformation of the response did help make the distribution of error more equal and normal than before.

We now check this new model's MAPE for accuracy:

```
(mape2 = kfold_cv(house_prices, house_prices.lm, 5, log_response = TRUE))
```

```
## [1] 11.94528
```

We see that this model's error of 11.9452845% has improved slightly relative to the original (non-log-transformed) version.

Significance of predictors

We check the model for the significance of the predictors by doing a t-test of each parameter, using an $\alpha = 0.10$ for significance.

```
summary(house_prices.lm)
```

```
##
## Call:
## lm(formula = log(SalePrice) ~ GrLivArea + LotArea + GarageArea +
##     BedroomAbvGr + FullBath + HalfBath + TotalBsmtSF + OverallQual +
##     OverallCond + YearBuilt + BldgType + HouseStyle + CentralAir +
##     Heating, data = house_prices)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.13813 -0.07149  0.00254  0.07319  0.49384
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.530e+00  6.011e-01   5.873 5.71e-09 ***
## GrLivArea         2.514e-04  2.159e-05  11.647  < 2e-16 ***
## LotArea           2.228e-06  6.170e-07   3.611 0.000319 ***
## GarageArea        2.144e-04  3.115e-05   6.882 1.00e-11 ***
## BedroomAbvGr     -3.184e-03  8.217e-03  -0.388 0.698451
## FullBath          4.330e-02  1.437e-02   3.013 0.002651 **
## HalfBath          3.570e-02  1.419e-02   2.515 0.012052 *
## TotalBsmtSF       5.205e-05  1.899e-05   2.741 0.006230 **
## OverallQual       9.496e-02  6.066e-03  15.655  < 2e-16 ***
## OverallCond       5.240e-02  5.124e-03  10.226  < 2e-16 ***
## YearBuilt         3.513e-03  3.133e-04  11.212  < 2e-16 ***
## BldgType2fmCon    1.296e-02  3.724e-02   0.348 0.727924
## BldgTypeDuplex   -1.119e-01  2.891e-02  -3.871 0.000115 ***
## BldgTypeTwnhs    -1.737e-01  2.934e-02  -5.921 4.32e-09 ***
## BldgTypeTwnhsE   -5.714e-02  2.017e-02  -2.833 0.004704 **
## HouseStyle1.5Unf  3.929e-02  6.214e-02   0.632 0.527331
## HouseStyle1Story  4.940e-02  2.109e-02   2.343 0.019329 *
## HouseStyle2.5Fin -3.957e-02  6.629e-02  -0.597 0.550629
## HouseStyle2.5Unf  1.534e-04  5.392e-02   0.003 0.997731
## HouseStyle2Story -3.325e-02  2.106e-02  -1.579 0.114739
## HouseStyleSFoyer  4.216e-02  3.714e-02   1.135 0.256608
## HouseStyleSLvl    2.796e-02  2.792e-02   1.001 0.316951
## CentralAirY       6.352e-02  2.692e-02   2.360 0.018477 *
## HeatingGasW       3.178e-02  4.923e-02   0.646 0.518716
## HeatingGrav      -7.179e-02  7.207e-02  -0.996 0.319405
## HeatingOthW      -7.618e-02  1.183e-01  -0.644 0.519833
## HeatingWall       9.365e-02  8.643e-02   1.084 0.278811
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1592 on 1068 degrees of freedom
## Multiple R-squared:  0.8389, Adjusted R-squared:  0.835
## F-statistic:   214 on 26 and 1068 DF,  p-value: < 2.2e-16
```

We see that there are numerous β-parameters with high p-values, such as for example `BedroomAbvGr`, `Heating`, and `HouseStyle`. We attempt to remove these and see if our model improves.

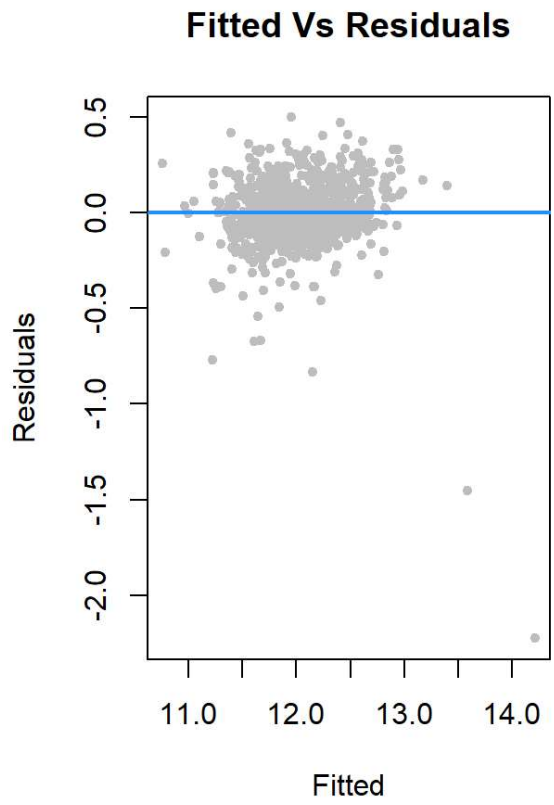
```
house_prices.lm = lm(log(SalePrice) ~ GrLivArea + LotArea + GarageArea + FullBath + HalfBath + TotalBsmt
SF + OverallQual + OverallCond + YearBuilt + BldgType + CentralAir, data = house_prices)
summary(house_prices.lm)
```

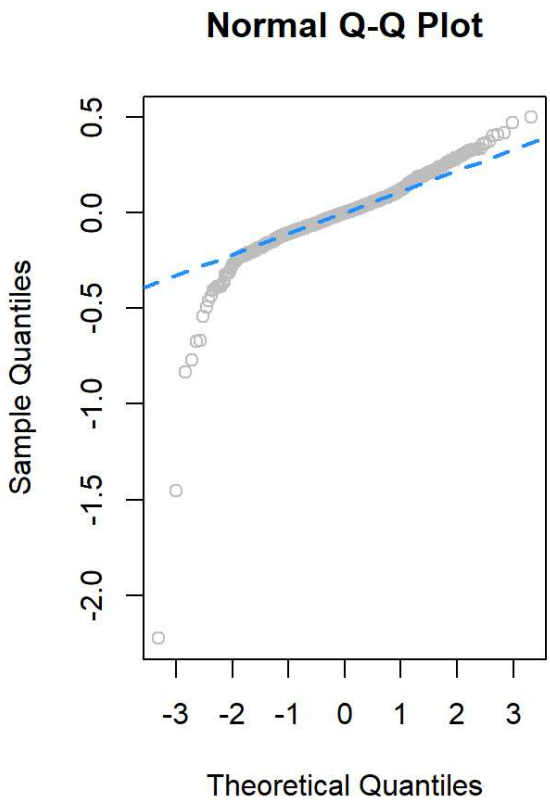
```
##
## Call:
## lm(formula = log(SalePrice) ~ GrLivArea + LotArea + GarageArea +
##     FullBath + HalfBath + TotalBsmtSF + OverallQual + OverallCond +
##     YearBuilt + BldgType + CentralAir, data = house_prices)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.22590 -0.07210  0.00162  0.07552  0.49934
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.462e+00  5.617e-01   6.165 9.96e-10 ***
## GrLivArea       2.108e-04  1.788e-05  11.792  < 2e-16 ***
## LotArea         2.288e-06  6.166e-07   3.711 0.000217 ***
## GarageArea      2.375e-04  3.044e-05   7.800 1.45e-14 ***
## FullBath        3.653e-02  1.388e-02   2.631 0.008627 **
## HalfBath        9.672e-03  1.257e-02   0.769 0.441845
## TotalBsmtSF     9.293e-05  1.536e-05   6.049 2.00e-09 ***
## OverallQual     9.220e-02  5.848e-03  15.766  < 2e-16 ***
## OverallCond     5.249e-02  5.082e-03  10.328  < 2e-16 ***
## YearBuilt       3.573e-03  2.913e-04  12.267  < 2e-16 ***
## BldgType2fmCon -3.680e-03  3.680e-02  -0.100 0.920355
## BldgTypeDuplex -8.885e-02  2.718e-02  -3.269 0.001115 **
## BldgTypeTwnhs  -1.903e-01  2.841e-02  -6.699 3.37e-11 ***
## BldgTypeTwnhsE -5.644e-02  1.923e-02  -2.935 0.003411 **
## CentralAirY     6.401e-02  2.421e-02   2.643 0.008325 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1599 on 1080 degrees of freedom
## Multiple R-squared:  0.8357, Adjusted R-squared:  0.8336
## F-statistic: 392.4 on 14 and 1080 DF,  p-value: < 2.2e-16
```

Now all of our β-parameters, apart from the dummy variable for `BldgType2fmCon` seem to be significant. Because the other dummy variables seem to be significant, we'll leave `BldgType` in.

Let's look at the Fitted Vs Residuals and the Q-Q Plot for this new model

```
diagnostics(house_prices.lm)
```





The variance looks consistent from left to right. When looking at the Q-Q Plot we can see that there seems to be some significant deviation from the line towards the left side of the graph.

We also check cross-validated MAPE:

```
(mape3 = kfold_cv(house_prices, house_prices.lm, 5, log_response = TRUE))
```

```
## [1] 11.69764
```

We see that the performance of our regression continues to improve, with a cross-validated MAPE of 11.6976414.

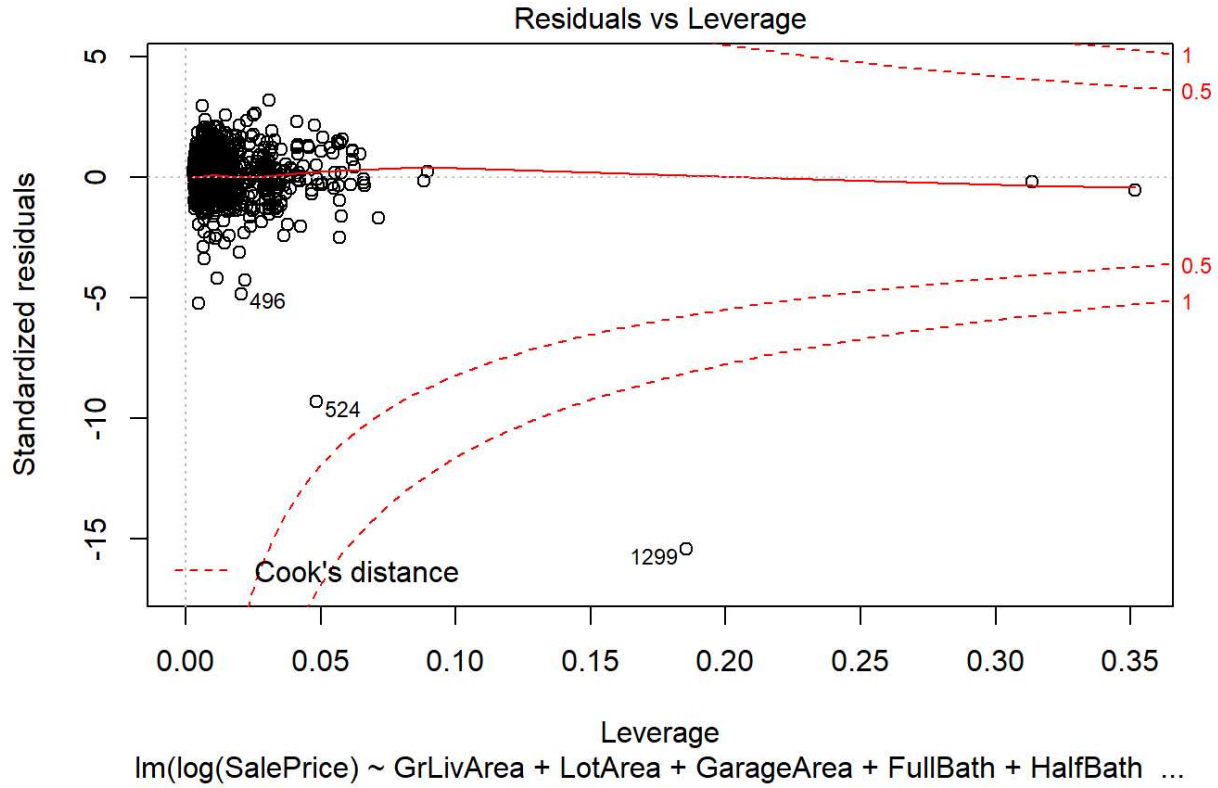
Analysis of outliers

We still have an issue with our model diagnostics, as the Normal Q-Q plot still shows substantial deviation from normal for our residuals.

OUr log response transformation has not worked to correct the problem and so we now turn to outliers.

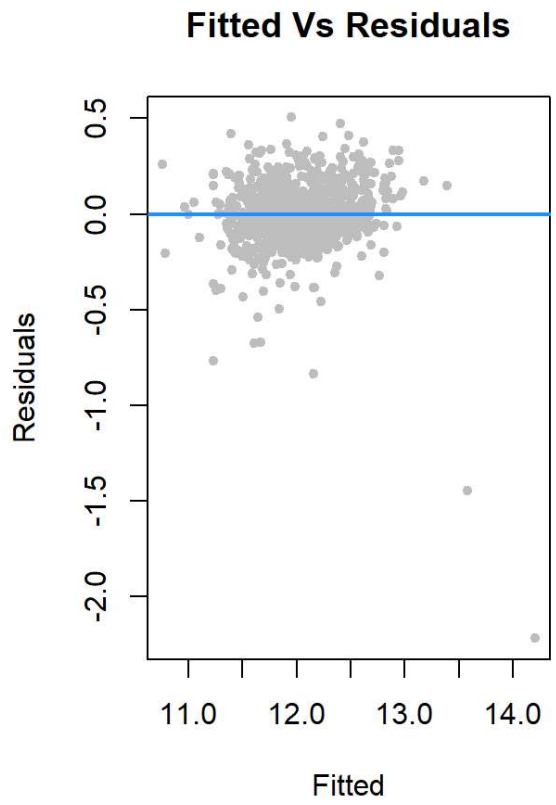
We examune a plot showing leverage, standardized residuals, and Cook's distance (red dotted lines).

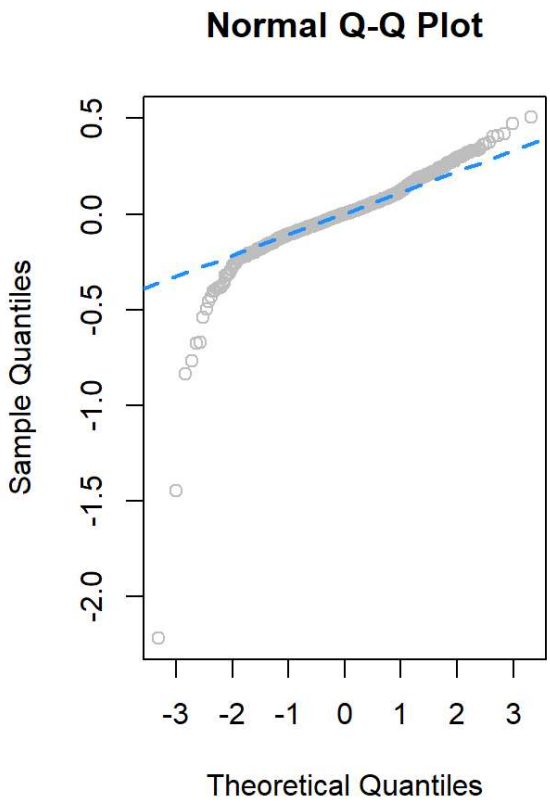
```
plot(house_prices.lm, which = 5)
```



There is one quite extreme outlier in observation #1299. Looking at the underlying data, we see that the data quality of this point is highly questionable. We conclude that this point can be safely removed from the data set.

```
influential_indices = as.vector(which(cooks.distance(house_prices.lm) > 4/length(cooks.distance(house_pr
ices.lm))))
house_prices_outliers_removed.lm = lm(log(SalePrice) ~ GrLivArea + LotArea + GarageArea + FullBath + Hal
fBath + TotalBsmtSF + OverallQual + OverallCond + YearBuilt + BldgType + CentralAir, data = house_prices
[-influential_indices[1], ])
diagnostics(house_prices_outliers_removed.lm)
```





Removing this one observation did improve the Normal Q-Q plot, but not enough to convince us that our residuals follow a normal distribution.

We check the new model accuracy:

```
(mape4 = kfold_cv(house_prices[-influential_indices[1],], house_prices_outliers_removed.lm, 547, log_res
ponse = TRUE))
```

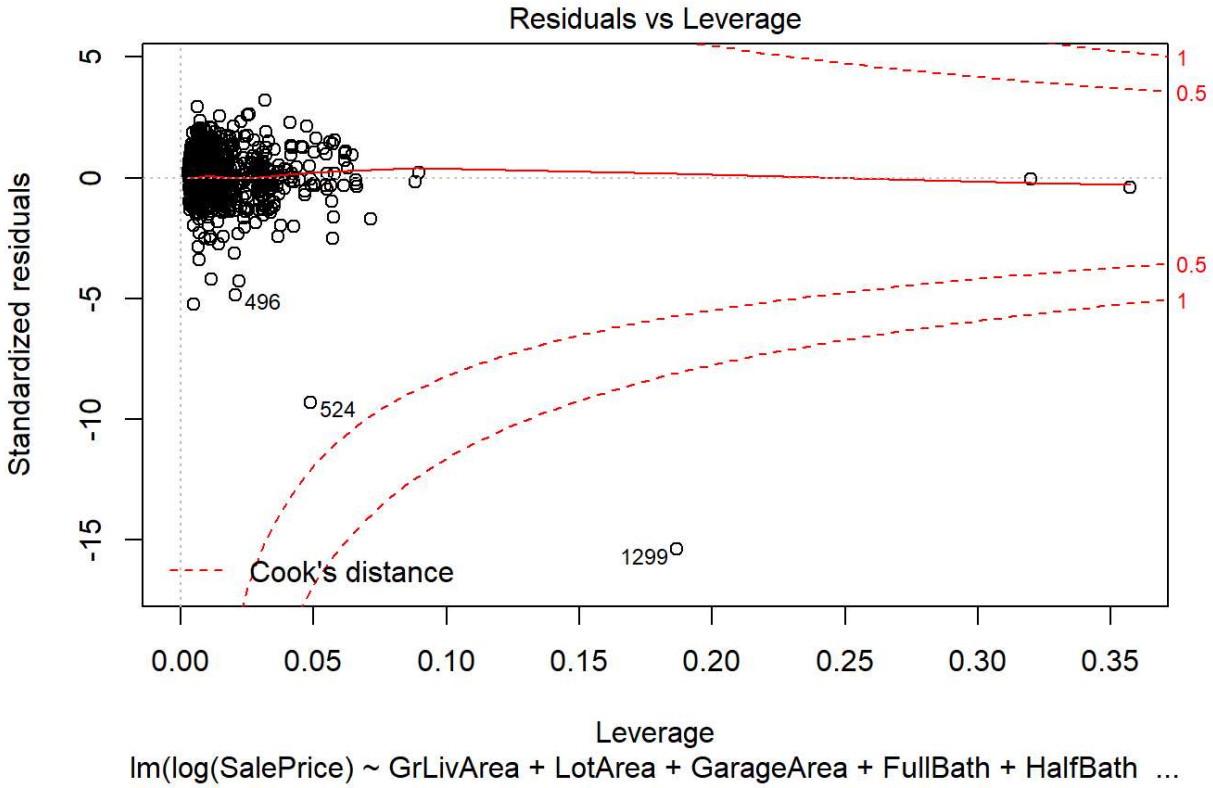
```
## [1] 11.82133
```

With the removal of a single bad observation we have improved the model performance substantially. We now have a MAPE of 11.8213331, exceeding our original goal for the project. We have a very good model for prediction performance.

Removal of outliers to correct violations of LINE assumptions

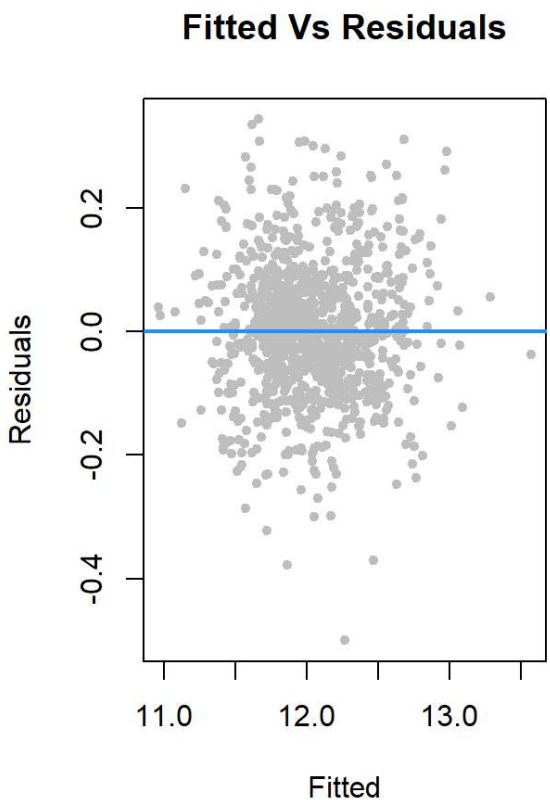
Our diagnostic plots are still not entirely satisfactory. Let's continue our analysis of outliers.

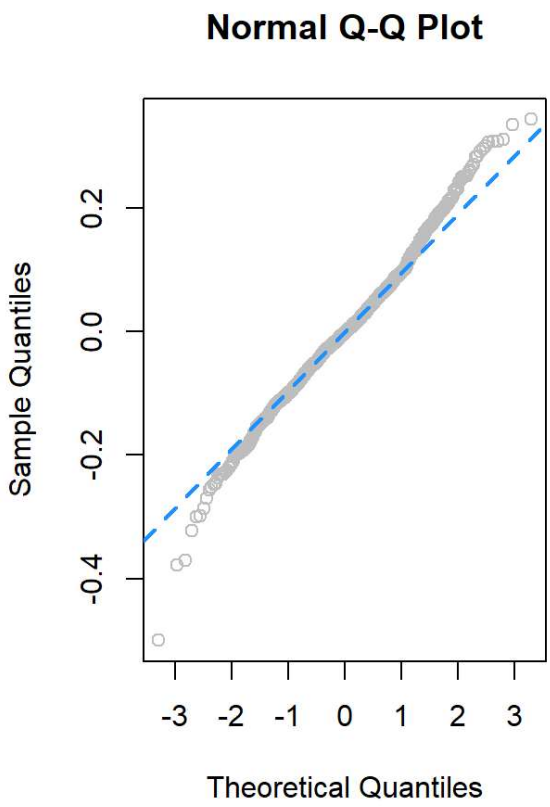
```
plot(house_prices_outliers_removed.lm, which = 5)
```



We see that there are still significant outliers in the model. We will now fit our model after removing all outliers (defined as those points have a Cook's distance of > 4/n).

```
influential_indices = as.vector(which(cooks.distance(house_prices.lm) > 4/length(cooks.distance(house_pr
ices.lm))))
house_prices_outliers_removed.lm = lm(log(SalePrice) ~ GrLivArea + LotArea + GarageArea + FullBath + Hal
fBath + TotalBsmtSF + OverallQual + OverallCond + YearBuilt + BldgType + CentralAir, data = house_prices
[-influential_indices, ])
diagnostics(house_prices_outliers_removed.lm)
```





The results are satisfying, at least visually. We have plots that give some confidence of having equal variance and normal distribution of error.

```
bptest(house_prices_outliers_removed.lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  house_prices_outliers_removed.lm
## BP = 73.959, df = 14, p-value = 3.671e-10
```

```
shapiro.test(resid(house_prices_outliers_removed.lm))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(house_prices_outliers_removed.lm)
## W = 0.99161, p-value = 1.286e-05
```

While we would still have to reject the null hypothesis with these two tests and conclude that it is unlikely that our error distribution is equal and normal, the performance is better than the unmodified original model.

We now look at how this model based on removing outliers performs using k-fold cross validation:

```
length(coef(house_prices_outliers_removed.lm))
```

```
## [1] 15
```

```
(mape5 = kfold_cv(house_prices[-influential_indices, ], house_prices_outliers_removed.lm, 6, log_respons
e = TRUE))
```

```
## Warning in split.default(sample(1:nrow(data)), 1:k): data length is not a
## multiple of split variable
```

```
## [1] 8.362371
```

Our k-fold cross-validation shows that performance on this model is very good, 8.362371%.

```
summary(house_prices_outliers_removed.lm)
```

```
##
## Call:
## lm(formula = log(SalePrice) ~ GrLivArea + LotArea + GarageArea +
##     FullBath + HalfBath + TotalBsmtSF + OverallQual + OverallCond +
##     YearBuilt + BldgType + CentralAir, data = house_prices[-influential_indices,
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.49994 -0.06512 -0.00222  0.06288  0.34305
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.011e+00  3.946e-01   7.631 5.34e-14 ***
## GrLivArea       2.725e-04  1.291e-05  21.110  < 2e-16 ***
## LotArea         3.283e-06  5.605e-07   5.858 6.32e-09 ***
## GarageArea      2.207e-04  2.205e-05  10.010  < 2e-16 ***
## FullBath        3.749e-03  1.012e-02   0.370  0.71115
## HalfBath        1.635e-03  8.697e-03   0.188  0.85088
## TotalBsmtSF     1.686e-04  1.136e-05  14.845  < 2e-16 ***
## OverallQual     7.691e-02  4.124e-03  18.651  < 2e-16 ***
## OverallCond     5.328e-02  3.629e-03  14.682  < 2e-16 ***
## YearBuilt       3.797e-03  2.050e-04  18.527  < 2e-16 ***
## BldgType2fmCon -3.330e-02  3.353e-02  -0.993  0.32100
## BldgTypeDuplex -1.141e-01  2.130e-02  -5.355 1.06e-07 ***
## BldgTypeTwnhs  -1.575e-01  2.020e-02  -7.797 1.56e-14 ***
## BldgTypeTwnhsE -3.866e-02  1.335e-02  -2.895  0.00387 **
## CentralAirY     4.515e-02  1.847e-02   2.444  0.01468 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1075 on 1019 degrees of freedom
## Multiple R-squared:  0.9161, Adjusted R-squared:  0.9149
## F-statistic: 794.2 on 14 and 1019 DF,  p-value: < 2.2e-16
```

Simplifying the model

We are satisfied with model prediction performance, and so we check to see if the model can be simplified further using automated techniques like stepwise AIC and BIC automated model selection.

Backwards AIC

We start with our model and try backwards AIC.

```
house_prices_outliers_removed.lm = lm(log(SalePrice) ~ GrLivArea + LotArea + GarageArea + BedroomAbvGr +
FullBath + HalfBath + TotalBsmtSF + OverallQual + OverallCond + YearBuilt + BldgType + HouseStyle + Cent
ralAir + Heating, data = house_prices[-influential_indices,])
house_prices_outliers_removed.lm = step(house_prices_outliers_removed.lm, direction = "backward", trace
 = 0)
summary(house_prices_outliers_removed.lm)
```

```
##
## Call:
## lm(formula = log(SalePrice) ~ GrLivArea + LotArea + GarageArea +
##     BedroomAbvGr + TotalBsmtSF + OverallQual + OverallCond +
##     YearBuilt + BldgType + CentralAir + Heating, data = house_prices[-influential_indices,
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51584 -0.06309 -0.00132  0.06324  0.33611
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.900e+00  3.503e-01   8.280 3.86e-16 ***
## GrLivArea       2.981e-04  1.129e-05  26.390  < 2e-16 ***
## LotArea         3.254e-06  5.554e-07   5.858 6.32e-09 ***
## GarageArea      2.043e-04  2.221e-05   9.201  < 2e-16 ***
## BedroomAbvGr   -2.042e-02  5.783e-03  -3.532 0.000431 ***
## TotalBsmtSF     1.653e-04  1.045e-05  15.815  < 2e-16 ***
## OverallQual     7.545e-02  4.131e-03  18.265  < 2e-16 ***
## OverallCond     5.319e-02  3.616e-03  14.711  < 2e-16 ***
## YearBuilt       3.869e-03  1.812e-04  21.353  < 2e-16 ***
## BldgType2fmCon -1.789e-02  3.364e-02  -0.532 0.595095
## BldgTypeDuplex -1.177e-01  2.145e-02  -5.487 5.15e-08 ***
## BldgTypeTwnhs  -1.671e-01  2.015e-02  -8.290 3.57e-16 ***
## BldgTypeTwnhsE -5.342e-02  1.389e-02  -3.847 0.000127 ***
## CentralAirY     6.392e-02  2.028e-02   3.152 0.001671 **
## HeatingGasW     3.310e-02  3.432e-02   0.965 0.334947
## HeatingGrav     3.826e-04  5.167e-02   0.007 0.994094
## HeatingOthW    -5.043e-02  7.861e-02  -0.641 0.521349
## HeatingWall     1.780e-01  6.587e-02   2.703 0.006983 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1065 on 1016 degrees of freedom
## Multiple R-squared:  0.9178, Adjusted R-squared:  0.9164
## F-statistic: 667.3 on 17 and 1016 DF,  p-value: < 2.2e-16
```

```
length(coef(house_prices_outliers_removed.lm))
```

```
## [1] 18
```

```
(mape6 = kfold_cv(house_prices[-influential_indices,], house_prices_outliers_removed.lm, 6, log_response
= TRUE))
```

```
## Warning in split.default(sample(1:nrow(data)), 1:k): data length is not a
## multiple of split variable
```

```
## [1] 8.291547
```

We see that AIC performance is marginally better (8.291547) with one fewer predictor (full bath).

Backwards BIC

Similarly, we try backwards BIC.

```
house_prices_outliers_removed.lm = lm(log(SalePrice) ~ GrLivArea + LotArea + GarageArea + BedroomAbvGr +
FullBath + HalfBath + TotalBsmtSF + OverallQual + OverallCond + YearBuilt + BldgType + HouseStyle + Cent
ralAir + Heating, data = house_prices[-influential_indices,])
house_prices_outliers_removed.lm = step(house_prices_outliers_removed.lm, direction = "backward", k = lo
g(nrow(house_prices[-influential_indices,])), trace = 0)
summary(house_prices_outliers_removed.lm)
```

```
##
## Call:
## lm(formula = log(SalePrice) ~ GrLivArea + LotArea + GarageArea +
##     BedroomAbvGr + TotalBsmtSF + OverallQual + OverallCond +
##     YearBuilt + BldgType + CentralAir, data = house_prices[-influential_indices,
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51748 -0.06283 -0.00139  0.06294  0.33639
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     2.895e+00  3.448e-01   8.398  < 2e-16 ***
## GrLivArea       2.990e-04  1.130e-05  26.445  < 2e-16 ***
## LotArea         3.275e-06  5.563e-07   5.886 5.36e-09 ***
## GarageArea      2.078e-04  2.214e-05   9.389  < 2e-16 ***
## BedroomAbvGr   -2.076e-02  5.773e-03  -3.596 0.000339 ***
## TotalBsmtSF     1.628e-04  1.041e-05  15.630  < 2e-16 ***
## OverallQual     7.483e-02  4.118e-03  18.174  < 2e-16 ***
## OverallCond     5.281e-02  3.605e-03  14.648  < 2e-16 ***
## YearBuilt       3.881e-03  1.787e-04  21.718  < 2e-16 ***
## BldgType2fmCon -2.294e-02  3.340e-02  -0.687 0.492399
## BldgTypeDuplex -1.084e-01  2.114e-02  -5.126 3.53e-07 ***
## BldgTypeTwnhs  -1.674e-01  2.018e-02  -8.296 3.38e-16 ***
## BldgTypeTwnhsE -5.357e-02  1.389e-02  -3.855 0.000123 ***
## CentralAirY     5.174e-02  1.832e-02   2.825 0.004822 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1068 on 1020 degrees of freedom
## Multiple R-squared:  0.9171, Adjusted R-squared:  0.916
## F-statistic: 867.9 on 13 and 1020 DF,  p-value: < 2.2e-16
```

```
length(coef(house_prices_outliers_removed.lm))
```

```
## [1] 14
```

```
(mape7 = kfold_cv(house_prices[-influential_indices,], house_prices_outliers_removed.lm, 6, log_response
= TRUE))
```

```
## Warning in split.default(sample(1:nrow(data)), 1:k): data length is not a
## multiple of split variable
```

```
## [1] 8.296839
```

We see that the resulting model performs well (8.2968394 - only fractionally better than our originally proposed model), and uses one less predictor (full bath) than our original model.

This is identical to the AIC-based model, above.

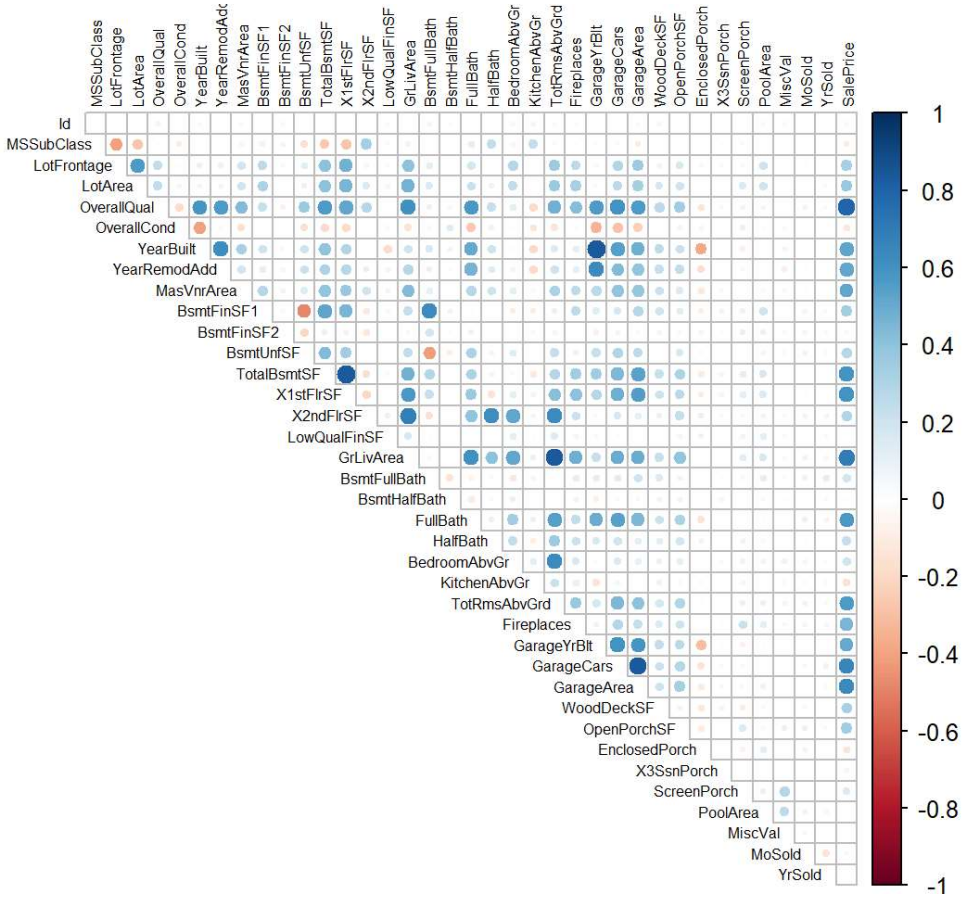
Analysis of correlation

The exclusion of “full bath” from our new model is not suprising when you observe the t-tests for full bath in the preceding sections. This does suggest that the variable is insignificant. We look at a plot of correlation to see if it brings insight into the data.

```
X_numeric = na.omit(house_prices[sapply(house_prices, is.numeric)])
correlation = round(cor(X_numeric, use = "everything"), 4)
```

Now plot the correlations:

```
corrplot(correlation, type = "upper", tl.pos = "td",
         method = "circle", tl.cex = 0.5, tl.col = 'black',
         diag = FALSE)
```



Indeed, we see here that the number of full baths is strongly positively correlated with with general living area and overall quality of the home. Since both of those variables are already in our model, it is not surprising that full bath is somewhat superfluous.

Results

We have selected a final model and validated its performance and adherence to LINE assumptions. We finally demonstrate the performance of the model and its diagnostics.

Performance against held out 25% of test data

Let's see how this model performs for making predictions against our test data.

```
influential_indices = as.vector(which(cooks.distance(house_prices.lm) > 4/length(cooks.distance(house_pr
ices.lm))))
house_prices_outliers_removed.lm = lm(log(SalePrice) ~ GrLivArea + LotArea + GarageArea + FullBath + Hal
fBath + TotalBsmtSF + OverallQual + OverallCond + YearBuilt + BldgType + CentralAir, data = house_prices
[-influential_indices, ])
```

```
house_prices_prediction = exp(predict(house_prices_outliers_removed.lm,
                                       newdata = house_prices_test))
```

```
mape(actual = house_prices_test$SalePrice, house_prices_prediction)
```

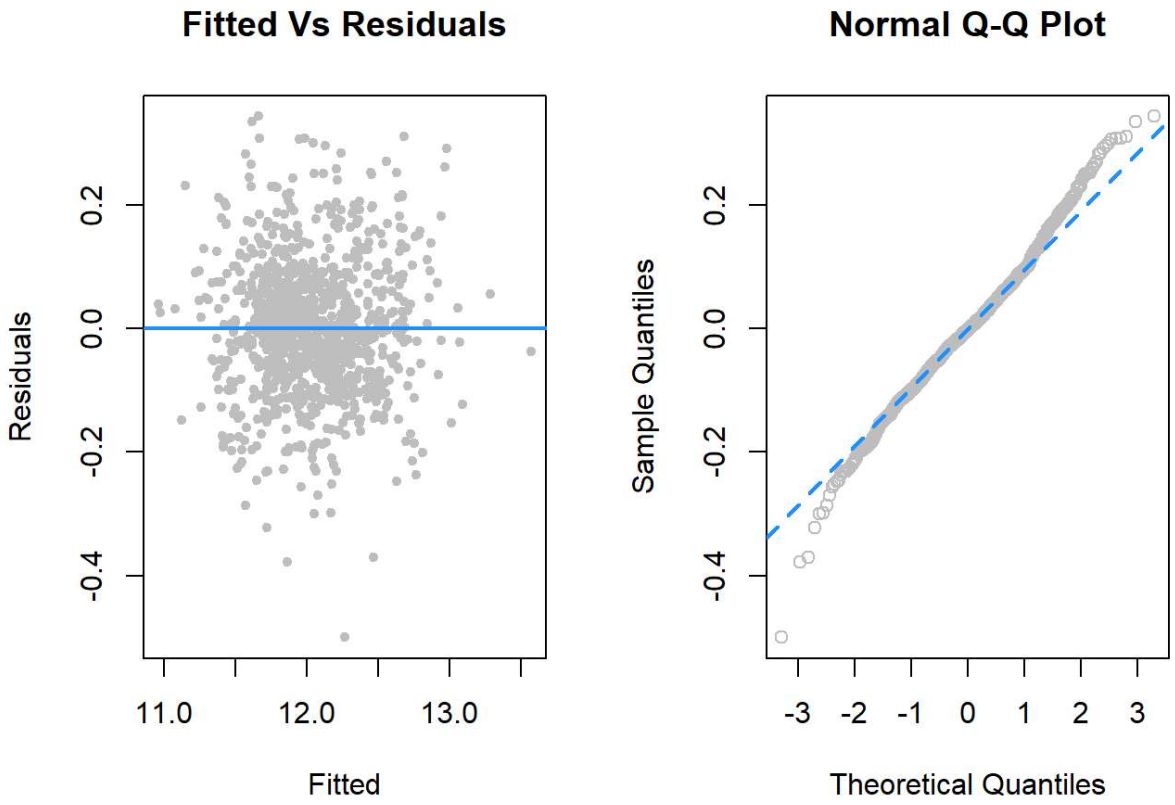
```
## [1] 10.60843
```

A 10.61% error falls within our target goal interval of 15% and ideally under 10%. We did not remove outliers from the test data and so it is not surprising that performance is slightly worse than we we saw during cross-validated testing of our training set.

Final model diagnostics (for LINE assumptions)

One of our other goal was for the the model to adhere to LINE assumptions. We can see in the Fitted Vs Residuals and Q-Q Plot that the model demonstrates both normality and relatively constant variance.

```
diagnostics(house_prices_outliers_removed.lm)
```



Adjusted R-squared

Adjusted R^2 value is:

```
get_adj_r2(house_prices_outliers_removed.lm)
```

```
## [1] 0.9148983
```

The model is excellent from this standpoint, with over 92% of the variance in responses being accounted for by our regression.

Collinearity

Finally, we wanted to ensure that our model was stable and did not contain collinearity.

We calculate the Variance Inflation Factor for our parameters.

```
vif(house_prices_outliers_removed.lm)
```

```
##      GrLivArea        LotArea     GarageArea       FullBath       HalfBath
##       3.720711       1.247962       1.767703       2.709619       1.732507
##    TotalBsmtSF    OverallQual    OverallCond      YearBuilt BldgType2fmCon
##       1.895945       2.705943       1.377567       3.138659       1.059470
## BldgTypeDuplex  BldgTypeTwnhs BldgTypeTwnhsE    CentralAirY
##       1.070334       1.128769       1.191084       1.325499
```

We can see that all of vif's are less than 5, indicating that multicollinearity should not be an issue with the model.

Discussion

Due to the overwhelming number of variables in this data set, we felt it was best to select a probable set of variables based on domain insight gleaned from study of real estate websites. We used statistical analysis to confirm our choices and to tune the model appropriately. The reasoning being this approach is simple: the things that a buyer values in a home are likely propotional to the cost. Real estate offices and websites have likely narrowed down the things that are important to buyers so that the buyers can find a home easily and make a purchase through them.

Since all of the houses in this data set came from the same geographic area (Ames, Iowa), we can assume that the physical characteristics of the house determined its price, as opposed to the comparison of prices being driven by the housing market in which it resides (i.e. comparing prices in completely different markets, such as, South Dakota and southern California).

Our approach was incremental, starting with a proposed model and then using cross-validation and model diagnostics to continually refine the model and improve its characteristics.

As we could see from the small average percent error and high R^2 of our model, the characteristics most prominently featured on a real estate website were highly correlated with the price of the home.

In conclusion, the model performs well across multiple metrics, adhere adequately to LINE assumptions, and makes quite accurate predictions. This gives evidence to support the hypothesis that a good model can be made from the most prominent features displayed on real estate websites.