Andreas Tsoumpariotis (48297890), CJ Olson (47574869), and Aiden Berry (48356000)

A. Introduction

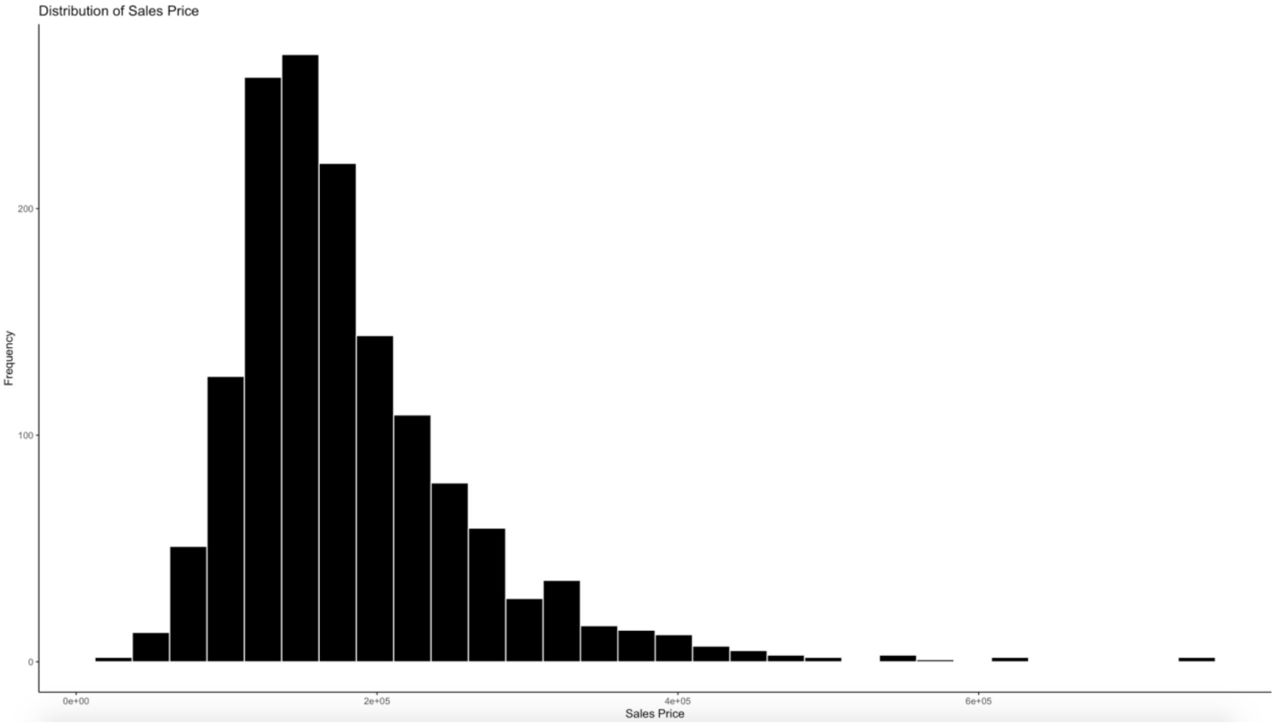
We were tasked with finding the answers to two analyses. First, we wanted to determine if the prices of houses are related to size and the neighborhood they are located in while considering three specific neighborhoods. To determine the answer to that, we created a multiple linear regression model. The point of this question is to consider the specific areas of interest and see how square footage impacts sales price for each neighborhood, and by how much on average.

Next, we wanted to determine what the most predictive model in terms of sales price of homes would be for all of the neighborhoods. We are confident that through our use of various variable selection processes, we came up with very strong models in terms of their ability to predict sales prices. The point of this question is to make the most comprehensive and complete models through the use of variable selection and data cleaning processes. Multiple models were created and a combination of factors including Kaggle score, RMSE, and statistical intuition, is how we determined which of the models is the best fit for our final proposed model.

B. Data Description

The original data set contains a total list of 2919 homes in Ames Iowa and 80 explanatory variables pertaining to features of the house, as well as the response variable *SalePrice* (81 variables total). 20 of the explanatory variables are continuous measurement variables relating to the dimensions of the house. 14 of the explanatory variables are discrete numeric variables that count the numbers of items in a house, mostly focussing on the number of rooms in the house (bathrooms, bedrooms, ect.). The rest are categorical variables ranging anywhere from 2 to 28 classes. The dataset was provided by Dean De Cock from Truman State University. The following link can be used to access the data as well as provide more in-depth descriptions about the variables and dataset as a whole: https://www.kaggle.com/c/house-prices-advanced-regression-techniques.

C. Data Cleaning



i) Data Cleaning for Analysis question 1

The parameters were pre-set for this question and we were only working with the train set, so we were limited to what we could do for our data cleaning procedures. Our client was only interested in three particular neighborhoods in Ames, so we filtered our data by these three neighborhoods: North Ames, Edwards, and Brookside. Next, since we know that our client prefers to talk about living areas in increments of 100 square feet, we made a new variable called *GrLivArea2* that accounts for this in our model (*GrLivArea2* = *GrLivArea*/100). Finally, we removed 15 observation points that had studentized residual values outside of the -2 to 2 range and opted against transforming the sales price variable for better interpretability of our results.

ii) Data Cleaning for Analysis question 2

Figure 2

Variable	Alley	PoolQC	Fence	MiscFeature	FireplaceQu
# of NAs	2721	2909	2348	2814	1420

For this question, our data cleaning procedures were much more in-depth and creative since we were working with the entire dataset of 81 variables and 2919 observations. We first combined the train and test datasets and dropped the *Id* and *SalePrice* variables (but will reinstate *SalePrice* again later). We used the *sapply()* function in R and found that *Alley*, *PoolQC*, *Fence*, *MiscFeature*, and *FireplaceQu* had large numbers of missing values (represented in *Figure 2*), so we dropped those variables as well. Next, we filtered our remaining variables as either numeric or categorical and imputed our numeric variables using mean imputation, and our categorical variables by mode imputation, in order to fully deal with any missing values. From this point, we decided to remove 27 more variables that didn't tell us much about *SalePrice*, such as random measurements, variables that had values that were mostly the same, years that things were built, etc.. Such variables include (a full disclosure of all variables can be seen within our code): *YearBuilt*, *LowQualFinSF*, *GarageYrBlt*, *Street*, *Utilities*, and *PavedDrive*. We made our remaining 47 variables numeric and noticed that both the *LotFrontage* and *LotArea* variables needed to be standardized, and we also reinstated *SalePrice* into our dataset again (it was removed prior to help with other data cleaning procedures), and this time standardizing it by taking its log (we can see in *Figure 1* how *SalePrice* needed to be standardized as it wasn't normally distributed). We set our boundaries a little more loosely for our studentized residuals (range was from -3 to 3) and removed 19 influential observations from our final train dataset.

D. Analysis Question 1

i) Problem**:**
Are the prices of houses related to the square footage of the living area of a house in the three different neighborhoods that our client services (Brookside, North Ames, and Edwards)?

ii) Model:

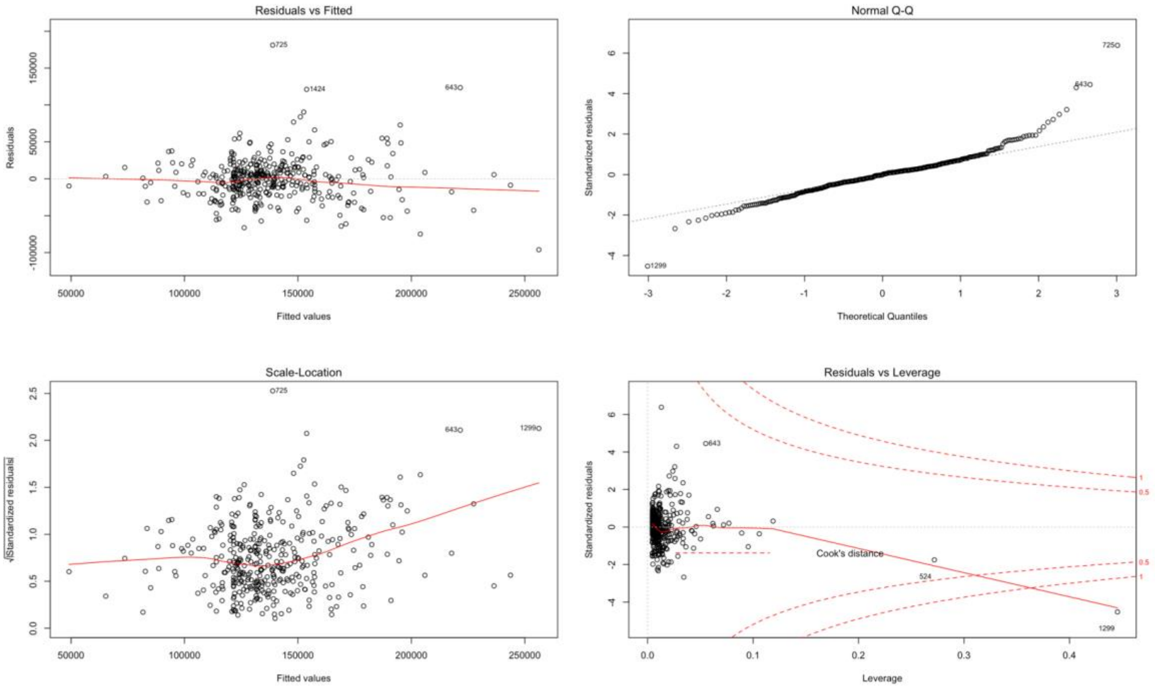
$$\widehat{SalePrice} = \beta_0 + \beta_1\text{GrLivArea2} + \beta_2\text{Neighborhood} + \beta_3(\text{GrLivArea2}\cdot\text{Neighborhood})$$

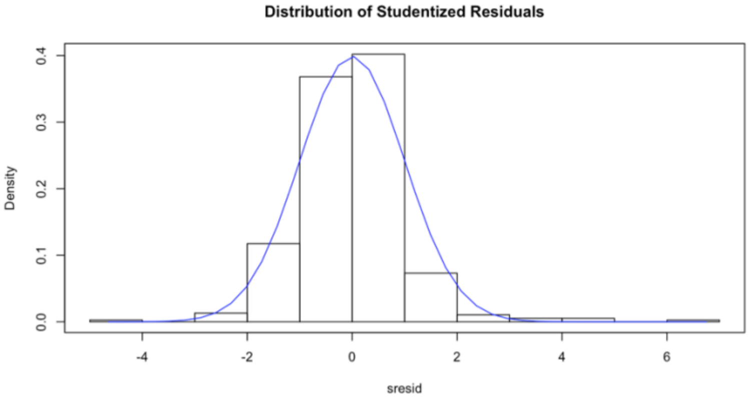
This model will help us discover if the square footage of the living area of a house in the three different neighborhoods is related to house sales prices. We decided to include an interaction term as we found it to be statistically significant with an extra sum of squares test (p-value < 0.01).

iii) Diagnostics:

Figure 3

a. Residual Plots





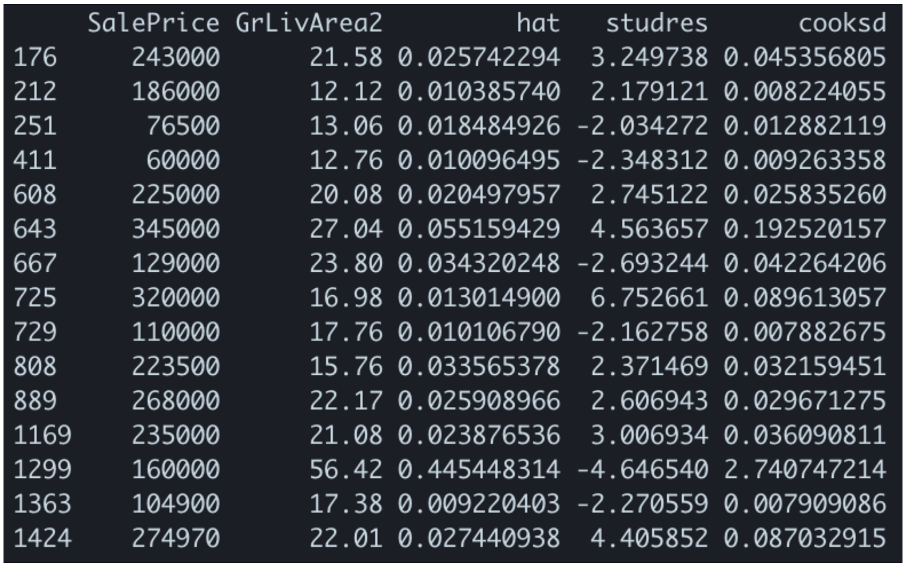
Normality: Judging from scatter plot, QQ plot, and histogram of residuals in *Figure 3*, there isn't strong evidence against the normality assumption, however there is minor evidence of outliers in the QQ plot, which we can see from the histogram as well since it has a slight right skew.
Linear Trend: There is a linear relationship between sales price and our independent variables.
Equal SD: There is little evidence from the scatter plots of heteroscedasticity.
Independence: We will assume the observations are independent.

b. Influential Point Analysis

We will proceed without taking any transformations as the diagnostic plots seem to be in good shape. The QQ plot and histogram does show some evidence of outliers, particularly a slight right skew. However, the dataset is quite large, so we aren't overly concerned about this. Also, there appears to be at least one influential point judging from the diagnostic plot in *Figure 3* (point 1299).

As seen in *Figure 4* below, 15 of the studentized residuals fall outside of the -2 to 2 range. Studentized residual values outside of the -2 to 2 range are good indicators of values that should be removed for being overly influential. We'll remove these 15 points and fit a model where they're not included.

Figure 4



```
     SalePrice GrLivArea2         hat   studres      cooksd
176     243000      21.58 0.025742294  3.249738 0.045356805
212     186000      12.12 0.010385740  2.179121 0.008224055
251      76500      13.06 0.018484926 -2.034272 0.012882119
411      60000      12.76 0.010096495 -2.348312 0.009263358
608     225000      20.08 0.020497957  2.745122 0.025835260
643     345000      27.04 0.055159429  4.563657 0.192520157
667     129000      23.80 0.034320248 -2.693244 0.042264206
725     320000      16.98 0.013014900  6.752661 0.089613057
729     110000      17.76 0.010106790 -2.162758 0.007882675
808     223500      15.76 0.033565378  2.371469 0.032159451
889     268000      22.17 0.025908966  2.606943 0.029671275
1169    235000      21.08 0.023876536  3.006934 0.036090811
1299    160000      56.42 0.445448314 -4.646540 2.740747214
1363    104900      17.38 0.009220403 -2.270559 0.007909086
1424    274970      22.01 0.027440938  4.405852 0.087032915
```

iv) Comparing Competing Models

$\widehat{SalePrice}_{BrkSide}$ = 22504.9 + (8490.9)·GrLivArea
$\widehat{SalePrice}_{Edwards}$ = 82146.3 + (3077.4)·GrLivArea
$\widehat{SalePrice}_{NAmes}$ = 78577 + (5130.8)·GrLivArea

The above models can be seen with the removed "problematic" points and the models are fit with the different neighborhoods set as the reference points.

v) Parameter Interpretation

Figure 5

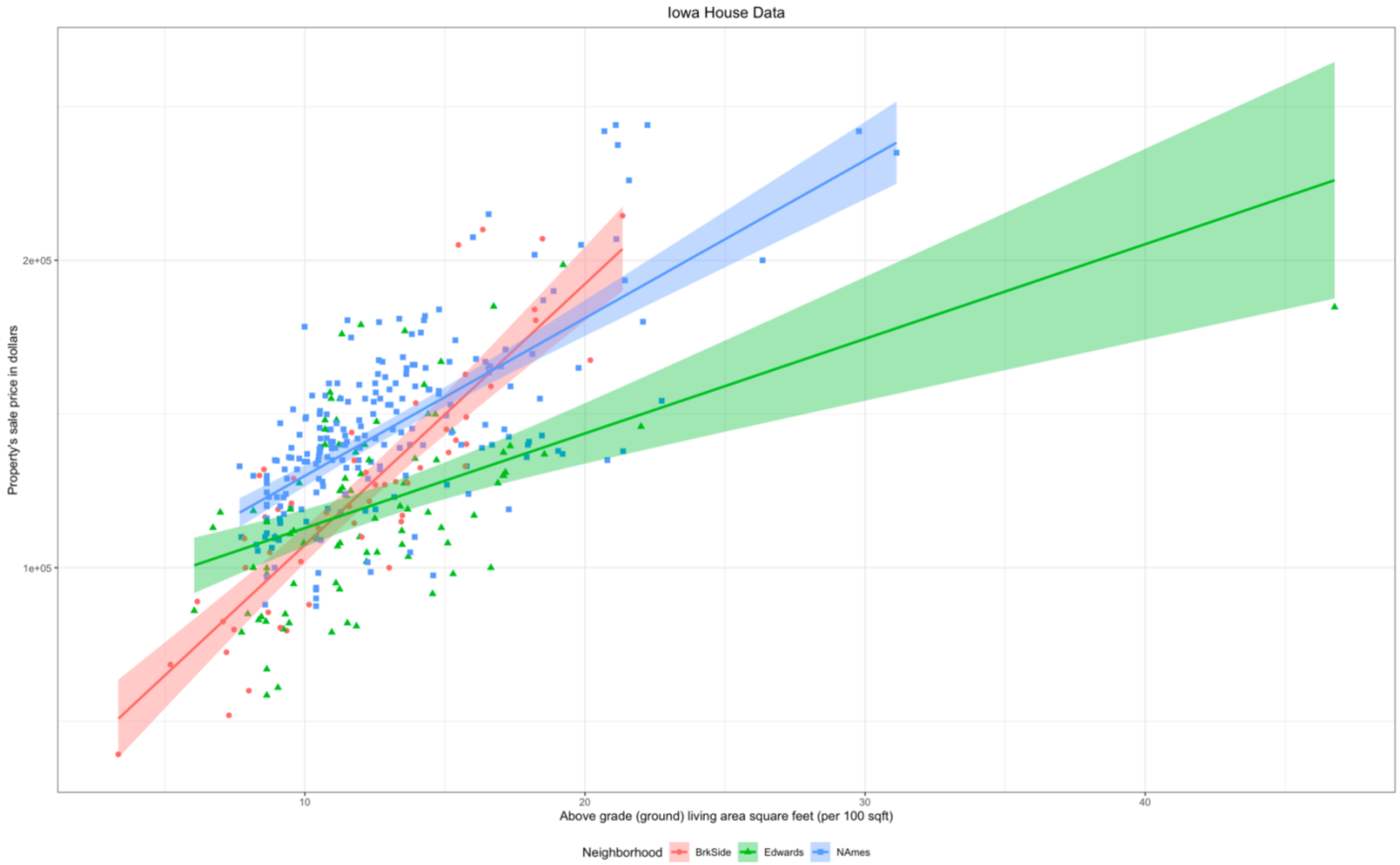
Neighborhood	Sales Price	Lower CI	Upper CI
Brookside	8,490.90	6,987.61	9,994.27
Edwards	3,077.40	2,123.35	4,031.38
North Ames	5,130.80	4,384.08	5,877.53

Figure 5 displays sales price and the confidence intervals for each of the three neighborhoods. With all other variables held constant, we can expect that an increase of 100 square feet of the living area of a house within the Brookside neighborhood is associated with an average increase of $8,490.90 of the sales price of the house. We are 95% confident that this increase of 100 square feet leads to an average increase in sales price between $6,987.61 and $9,994.27. For homes in the Edwards neighborhood, an increase of 100 square feet of the living area of a house is associated with an average increase of $3,077.40 of the sales price of the house. We are 95% confident that this average increase in sales price (for homes in the Edwards neighborhood) is between $2,123.35 and $4,031.38. Lastly, for homes in the North Ames neighborhood, an increase of 100 square feet of the living area of a house is associated with an average increase of $5,130.80 of the sales price of the house and we are 95% confident that this average increase in sales price (for homes in the North Ames neighborhood) is between $4,384.08 and $5,877.53.

vi) Conclusion

a) We can conclude that the neighborhood that a home is located in, as well as the square footage of the living area of the house, play important roles in deciding a home's sales price. Essentially, homes with the same square footage will have different sales prices depending on which neighborhood they're located in. Brookside homes will have greater average sales prices than North Ames and Edwards homes for homes that are the exact same size (in terms of square footage), even though North Ames has several of the most expensive houses of these three neighborhoods. For every 100 square foot increase of the living area of a house, we can see that homes in the Brookside neighborhood have a greater average increase in sales prices than homes that are in both the Edwards and North Ames neighborhoods, and homes in the North Ames neighborhood have a greater average increase in sales prices than homes that are in the Edwards neighborhood.

b) *Figure 6* below helps portray this story. The x-axis shows square footage (per 100 square feet) and the y-axis shows sales prices. The red points depict homes within the Brookside neighborhood, the blue are homes within the North Ames neighborhood, and the green are homes within the Edwards neighborhood. As we can see, the slope for the Brookside neighborhood homes is steeper than the slopes for the North Ames and Edwards neighborhood homes, and the slope for the North Ames neighborhood homes is steeper than the slope for the Edwards neighborhood homes, signifying the different average increases of sales prices as it pertains to the three given neighborhoods and square footage.



E. Analysis Question 2

i) Problem

Which model is the most successful in predicting sales prices for homes in all of Ames, Iowa? Essentially, we are trying to see which variables are the best in determining the sales prices of homes in Ames, utilizing all neighborhoods this time.

ii) Model Selection

a) Candidate variables

Final Predictors

$$\widehat{SalePrice} = \beta_0 + \beta_1\text{LotArea} + \beta_2\text{OverallQual} + \beta_3\text{OverallCond} + \beta_4\text{BsmtFinSF1} + \beta_5\text{TotalBsmtSF} + \beta_6\text{GrLivArea} + \beta_7\text{GarageCars} + \beta_8\text{CentralAir} + \beta_9\text{GarageFinish} + \beta_{10}\text{SaleCondition}$$

We first used backwards, forward, and stepwise variable selection methods to help find the most optimal predictors for our model, with the essential goal to predict sales prices for homes in Ames, Iowa. By first using various data cleaning procedures, such as variable and observation manipulation, as well as standardization methods, we were able to use these variable selection techniques in order to build an effective model. We used 10-fold cross-validation and set a model limitation of no more than 10 predictors for all three variable selection methods, and all three yielded a model with 10 predictors to be the most significant. All three also found *OverallQual* and *GrLivArea* to be the two most significant predictors. This means that sales prices are greatly driven by the overall material and finish of the house (although this variable can be subjective in terms of how someone views quality of a house as opposed to someone else) as well as the above ground living area in square ft. We decided to use the predictors chosen from forward selection since it had the best RMSE.

b) Competing methods

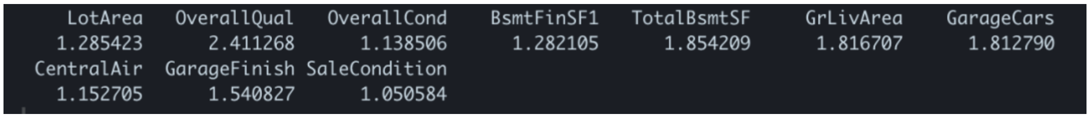
Backwards variable selection
Our backwards selection model chose *LotArea*, *OverallQual*, *OverallCond*, *BsmtUnfSF*, *TotalBsmtSF*, *GrLivArea*, *KitchenAbvGr*, *GarageCars*, *BsmtQual*, and *CentralAir* to be our predictors, with the two most significant being *OverallQual* and *GrLivArea*. Moreover, we obtained an RMSE of *0.1315878* for backwards selection.

Forward variable selection
Our forward selection model chose *LotArea*, *OverallQual*, *OverallCond*, *BsmtFinSF1*, *TotalBsmtSF*, *GrLivArea*, *GarageCars*, *BsmtQual*, *CentralAir*, *GarageFinish*, and *SaleCondition* to be our predictors, with the two most significant being *OverallQual* and *GrLivArea* as well. We obtained our best RMSE of *0.1292913* for forward selection.

Stepwise variable selection
Our stepwise selection model chose *MSSubClass*, *LotFrontage*, *LotArea*, *OverallQual*, *OverallCond*, *BsmtFinSF1*, *BsmtUnfSF*, *TotalBsmtSF*, *X1stFlrSF*, and *GrLivArea* to be our predictors, with the two most significant as *OverallQual* and *GrLivArea* once again. We obtained an RMSE of *0.1335056* for stepwise variable selection, which is our worst RMSE.

Penalized regression methods

Figure 7



```
    LotArea   OverallQual   OverallCond    BsmtFinSF1   TotalBsmtSF     GrLivArea    GarageCars
   1.285423      2.411268      1.138506      1.282105      1.854209      1.816707      1.812790
 CentralAir  GarageFinish SaleCondition
   1.152705      1.540827      1.050584
```

By utilizing penalized regression methods, such as ridge regression, lasso, and elastic net, we may be able to deal with potential multicollinearity issues within our data, and we may also obtain better RMSE values as well. We measured each predictor's VIF score in order to see their correlation with one another (seen in *Figure 7*). If predictors are highly correlated with each other, we may run into multicollinearity issues. Luckily, none of our predictors seem to be highly correlated with one another since we obtain low VIF scores. 10-fold cross-validation was again used for ridge regression and lasso, and repeated cross validation was used for the elastic net.

Ridge regression

Ridge regression introduces bias by shrinking the coefficient estimates in order to help against variables that may be highly correlated. Essentially, with ridge regression we can reduce our variance in order to obtain more accurate predictions, but we also have to keep in mind that we're introducing bias as well. When utilizing ridge regression, we obtain an RMSE of *0.1293589*.

Lasso

Lasso also shrinks the coefficient estimates, but can cause some to shrink to 0, and can essentially perform a bit of variable selection as well. For our data, lasso did not eliminate any more of our predictors. Also, we obtained a better RMSE of *0.1292841* for our lasso model.

Elastic Net

Elastic net has a mix between ridge and lasso components and includes tuning parameters from both. We obtained our best RMSE of *0.1265113* with the elastic net.

Final Model

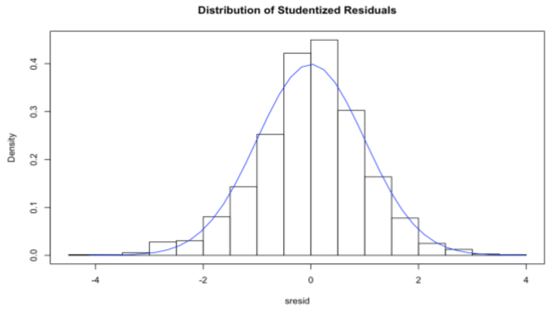
We used our forward model as our final multiple linear regression model since it obtained the best Kaggle score. Our **final model** is:

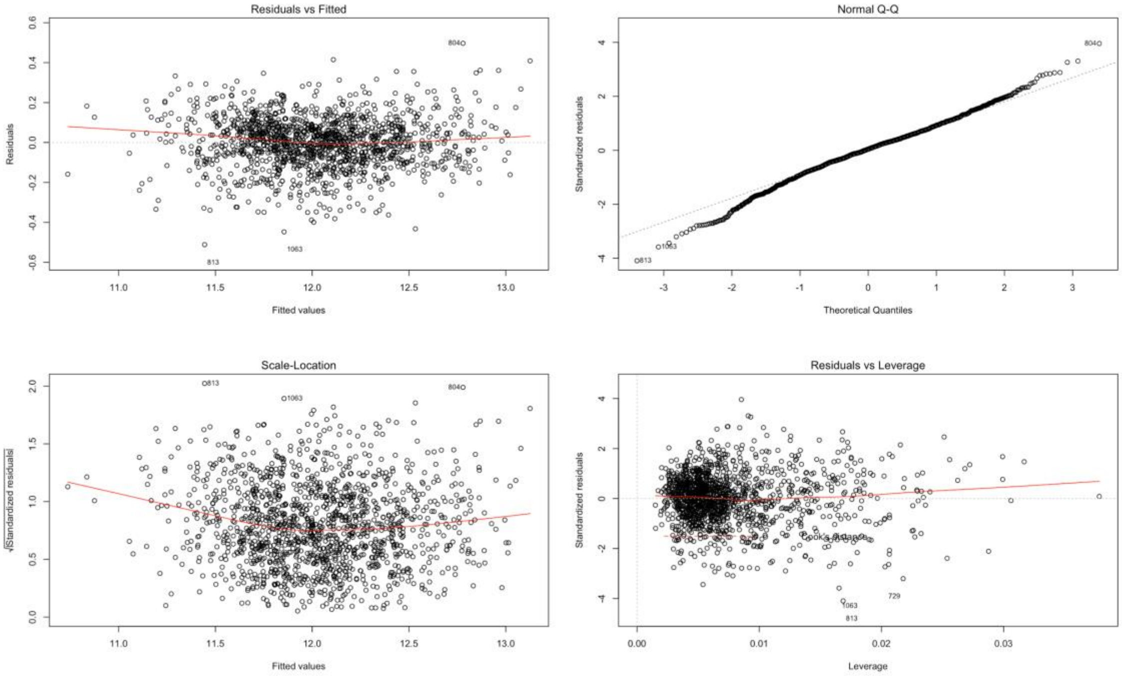
$$\widehat{SalePrice} = 10.1297 + (0.0369)\text{LotArea} + (0.1044)\text{OverallQual} + (\ 0.0323)\text{OverallCond} + (0.0002)\text{BsmtFinSF1} + (0.0002)\text{TotalBsmtSF} + (0.0003)\text{GrLivArea} + (0.0727)\text{GarageCars} + (0.1484)\text{CentralAir} + (-0.0462)\text{GarageFinish} + (0.0265)\text{SaleCondition}$$

iii) Addressing Diagnostics

Since our forward model obtained the best Kaggle score, we used it for our final diagnostics plots as well (seen below in *Figure 8*).

Figure 8





a) Residual plots

The scatter plot, QQ plot, histogram of residuals, and the fact that our dataset is also quite large, means that we aren’t concerned about normality issues. We get a nice scatter in our data and are quite confident about how our diagnostic plots represent our final model.

b) Influential point analysis (Cook’s D and Leverage)

Our final model doesn’t have any influential outliers, as seen from the Cook's distance plot right above, and it doesn’t seem to have observations with high leverage either. Values are below .04 on the x-axis, which indicates that we’re in good standing.

iv) Comparing Competing Models

The competing models, which are specified in this section of our analysis, can be seen in *Figure 9* below. We are comparing our models’ RMSE values and Kaggle scores. Moreover, we will not state our RMSE values and Kaggle scores for the backwards and stepwise selection methods since they obtained the worst scores and RMSE values.

Predictive Model	**RMSE**	**Kaggle Score**
Forward	0.1292913	0.14669
Ridge Regression	0.1293589	0.14877
Lasso	0.1292841	0.14681

Elastic Net	0.1265113	0.14692

Figure 9

Our forward model does not have the lowest RMSE, but obtained the best Kaggle score, while our elastic net model has the lowest lowest RMSE but obtained the second worst Kaggle score (of the ones represented in *Figure 9*). Nonetheless, our forward, ridge regression, lasso, and elastic net models all obtain Kaggle scores under .15, and despite our backwards and stepwise models obtaining the worst Kaggle scores, they still had quite low RMSE values as well.

v) Conclusion

In conclusion, our models did a great job in predicting sales prices for homes in Ames, Iowa. Our forward model obtained the best Kaggle score, and so we used it is our final model. Our two most significant predictors in predicting sales prices were *OverallQual* and *GrLivArea*, which were always selected from our utilized variable selection methods. Other variables, such as *TotalBsmtSF*, *GarageCars*, and *CentralAir* were consistently chosen as significant predictors by our models as well, which makes a lot of sense logically, since people tend to care about their basement and garage sizes as well as having central air conditioning. Penalized regression methods can help deal with multicollinearity issues within data, and so we were able to obtain even better predictions with those procedures compared to backwards and stepwise variable selection, but since our 10 selected variables did not have multicollinearity issues, our forward model still ended up being our best model. The predictors chosen in our forward model can definitely influence home sales prices as sales prices generally seem to be affected by things like overall home quality and condition, lot sizes, living area sizes, basement sizes, garage space and quality, having air conditioning, and the type of sale that is being made (normal, foreclosure, trade, etc.). One major trend that we noticed is that people are willing to spend more for homes that are of good quality and have a lot of space, maybe for storage and/or other things.

F. Code Appendix

```
### STAT 6301 Final Project ###

#Load packages
library(dplyr)
library(ggplot2)
library(gmodels)
library(agricolae)
library(multcomp)
library(Sleuth2)
library(MASS)
library(car)
library(glmnet)
library(caret)
library(leaps)
library(bestglm)
library(VIM)
library("VIM")
library(forcats)
library(stringr)
library(WriteXLS)

# Question 1 #

#House data (train set only)
house = read.csv("train.csv")
house = subset(house, Neighborhood=="NAmes" | Neighborhood=="Edwards" | Neighborhood=="BrkSide")
house$GrLivArea2 = house$GrLivArea/100
head(house)
str(house)

#Fit the model (ref is automatically set as 'BrkSide')
house.lm <- lm(SalePrice ~ GrLivArea2 + Neighborhood + GrLivArea2:Neighborhood, data = house)
summary(house.lm)

#Before proceeding, is the interaction term between "GrLivArea" and "Neighborhood" statistically significant?

#Re-fit model without interaction term
lm.without = lm(SalePrice ~ GrLivArea2 + Neighborhood, data = house)
summary(lm.without)
```

```
#Extra sum of squares test
anova(lm.without, house.lm)

# The extra sum of squares test is statistically significant (p-value < .01). That is, there's
sufficient
# evidence to conclude that any of the slope parameters related to combinations of square
footage of the
# living area of the house and the neighborhood that the house is located in are statistically
significant.

#Diagnostic plots of normal data
par(mfrow=c(2,2))
plot(house.lm)

#Histogram of normal data showing the distribution of the studentized residuals
sresid <- rstudent(house.lm)
hist(sresid, freq=FALSE, main="Distribution of Studentized Residuals")
box()
xfit <- seq(min(sresid), max(sresid), length=40)
yfit <- dnorm(xfit)
lines(xfit, yfit, col='blue')

#Scatterplot matrix
pairs(SalePrice ~ GrLivArea2 + Neighborhood + GrLivArea2:Neighborhood, data = house)

#Add leverage, studentized residuals, and Cook's D to data set
house = transform(house, hat = hatvalues(house.lm))
house = transform(house, studres = studres(house.lm))
house = transform(house, cooksd = cooks.distance(house.lm))

#Inspect and remove problematic points
h = subset(house, studres <= -2 | studres >= 2)
View(h) #see which observation points are problematic in order to remove (15 observation
points)
house = house %>% filter(Id != 176)
house = house %>% filter(Id != 212)
house = house %>% filter(Id != 251)
house = house %>% filter(Id != 411)
house = house %>% filter(Id != 608)
house = house %>% filter(Id != 643)
house = house %>% filter(Id != 667)
house = house %>% filter(Id != 725)
house = house %>% filter(Id != 729)
house = house %>% filter(Id != 808)
house = house %>% filter(Id != 889)
house = house %>% filter(Id != 1169)
house = house %>% filter(Id != 1299)
house = house %>% filter(Id != 1363)
house = house %>% filter(Id != 1424)

#Re-fit model without problematic points (ref is automatically set as 'BrkSide')
house.BrkSide <- lm(SalePrice ~ GrLivArea2 + Neighborhood + GrLivArea2:Neighborhood, data =
house)
summary(house.BrkSide)
confint(house.BrkSide)

#Check diagnostic plots and see if there are any more problematic points
par(mfrow=c(2,2))
plot(house.BrkSide)
h = subset(house, studres <= -2 | studres >= 2)
View(h) #there are no more problematic points

#Fit models with other neighborhoods as reference points

#Fit model (ref=Edwards)
house2 <- within(house, Neighborhood <- relevel(Neighborhood, ref = 'Edwards'))
house.Edwards <- lm(SalePrice ~ GrLivArea2 + Neighborhood + GrLivArea2*Neighborhood, data
= house2)
summary(house.Edwards)
confint(house.Edwards)

#Fit model (ref=NAmes)
house3 <- within(house, Neighborhood <- relevel(Neighborhood, ref = 'NAmes'))
house.NAmes <- lm(SalePrice ~ GrLivArea2 + Neighborhood + GrLivArea2*Neighborhood, data
= house3)
summary(house.NAmes)
confint(house.NAmes)

#Visual
#The crossed lines on the graph suggest that there is an interaction effect
ggplot(house, aes(x=GrLivArea2, y=SalePrice, shape=Neighborhood, color=Neighborhood)) +
  geom_point(size=2) +
  geom_smooth(method=lm, aes(fill=Neighborhood)) +
  ggtitle("Iowa House Data") +
  xlab("Above grade (ground) living area square feet (per 100 sqft)") +
  ylab("Property's sale price in dollars") +
  theme_bw() +
  theme(legend.position="bottom", plot.title = element_text(hjust = 0.5)) +
  labs(color='Neighborhood', shape='Neighborhood', fill='Neighborhood')

#Biggest increase of salesprice as it relates to GrLivArea (which has the steepest slope?)
# (1) BrkSide (2) NAmes (3) Edwards

# Question 2 #

#Read in dataset
train = read.csv("train.csv")
train = data.frame(train)
test = read.csv("test.csv")
test = data.frame(test)

#Histogram of SalesPrice
ggplot(train, aes(x=SalePrice)) + geom_histogram(color="white", fill="black") +
  labs(title="Distribution of Sales Price",x="Sales Price", y = "Frequency")+
  theme_classic() #not normally distributed

train.drop = train[,!(names(train) %in% c("Id","SalePrice"))]
test.drop = test[,!(names(test) %in% c("Id"))]
house.data = rbind(train.drop, test.drop)
str(house.data)

#See which variables have NA values
sapply(house.data, function(x) sum(is.na(x)))

#Variables with the greatest number of missing observations:
# Alley: 2721
# PoolQC: 2909
# Fence: 2348
# MiscFeature: 2814
# FireplaceQu: 1420

#Drop variables with large number of NA values
house.data = house.data[,!(names(house.data) %in%
c("Alley","FireplaceQu","PoolQC","Fence","MiscFeature"))]

#See which variables are numeric and impute by the mean
num = house.data[sapply(house.data,is.numeric)]
str(num)
num = kNN(num, k = 10, numFun = mean)
num = num[,-c(37:72)]

#See which variables are categorical and impute by mode (most frequently appearing value)
```

```
categorical = house.data[sapply(house.data,is.factor)]
str(categorical)
categorical = kNN(categorical, k = 10, numFun = mode)
categorical = categorical[,-c(39:76)]

#Combine again and see if any more NAs
house.data = cbind(num, categorical)
sapply(house.data, function(x) sum(is.na(x))) #no more NA values

#Remove more variables we don't need from the rest (i.e. random measurements, too many of
the same values,
# years things were built, anything that wouldn't tell us much about 'SalePrice', etc.)

var = colnames(house.data)
remove = c('YearBuilt', 'YearRemodAdd', 'MasVnrArea', 'BsmtFinSF2', 'X2ndFlrSF',
'LowQualFinSF', 'GarageYrBlt', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', 'X3SsnPorch',
'ScreenPorch', 'PoolArea', 'MiscVal', 'YrSold', 'Street', 'LandContour', 'Utilities', 'LandSlope',
'Exterior1st', 'Exterior2nd',  'Foundation', 'Heating', 'Electrical', 'GarageType', 'GarageCond',
'PavedDrive')
var = setdiff(var, remove)
house.data = house.data[, var]

#Make all variables numeric
house.data = data.frame(lapply(house.data, function(x) as.numeric(as.factor(x))))
str(house.data) #there are now 47 variables

#Train and test
set.seed(100)
train.index = nrow(train)
test.index = train.index + 1
total = train.index + nrow(test)

#log variables
house.data$LotFrontage = log(house.data$LotFrontage)
house.data$LotArea = log(house.data$LotArea)

final.train = house.data[1:train.index, ]
final.test = house.data[test.index:total, ]

final.train$SalePrice = log(train$SalePrice)

#Diagnostic plots
lm = lm(SalePrice ~ ., data=final.train)
par(mfrow=c(2,2))
```

```
plot(lm)

#Add leverage, studentized residuals, and Cook's D to data set
final.train = transform(final.train, hat = hatvalues(lm))
final.train = transform(final.train, studres = studres(lm))
final.train = transform(final.train, cooksd = cooks.distance(lm))

#Inspect and remove problematic points and hat, studres, and cooksd at the end
h = subset(final.train, studres <= -3 | studres >= 3)
View(h)
final.train = final.train[-
c(31,411,463,496,524,589,633,667,682,689,715,875,917,969,971,1183,1299,1325,1433),]
final.train = final.train[,!(names(final.train) %in% c("hat","studres","cooksd"))]

#Specify 10-fold cross-validation
train.control <- trainControl(method = "cv", number = 10)

#Backwards variable selection
back.model <- train(SalePrice ~ ., data = final.train,
              method = "leapBackward",
              tuneGrid = data.frame(nvmax = 1:10),
              trControl = train.control)

#Identify the optimal tuning parameter
back.model$bestTune
#Look at all results from tuning process
back.model$results[10,] #RMSE
#View results of the selection process for the optimal parameter
summary(back.model$finalModel)
#Look at coefficients
coef(back.model$finalModel, 10)
#Predict
back.pred = predict(back.model, as.matrix(final.test), s=back.model$bestTune)

#Forward variable selection
forward.model <- train(SalePrice ~ ., data = final.train,
                method = "leapForward",
                tuneGrid = data.frame(nvmax = 1:10),
                trControl = train.control)

forward.model$bestTune
forward.model$results[10,] #RMSE
summary(forward.model$finalModel)
coef(forward.model$finalModel, 10)
```

```
#Predict
forward.pred = predict(forward.model, as.matrix(final.test), s=forward.model$bestTune)

#Stepwise variable selection
step.model <- train(SalePrice ~ ., data = final.train,
                method = "leapSeq",
                tuneGrid = data.frame(nvmax = 1:10),
                trControl = train.control)

step.model$bestTune
step.model$results[10,] #RMSE
summary(step.model$finalModel)
coef(step.model$finalModel, 10)
#Predict
step.pred = predict(step.model, as.matrix(final.test), s=step.model$bestTune)

#Based off of variables chosen from forward selection (because lowest RMSE)
final.lm = lm(SalePrice ~
LotArea+OverallQual+OverallCond+BsmtFinSF1+TotalBsmtSF+GrLivArea+GarageCars+CentralAir
+GarageFinish+SaleCondition, data=final.train)
summary(final.lm)

final.train = final.train %>%
dplyr::select(LotArea,OverallQual,OverallCond,BsmtFinSF1,TotalBsmtSF,GrLivArea,GarageCars,C
entralAir,GarageFinish,SaleCondition,SalePrice)
final.test = final.test %>%
dplyr::select(LotArea,OverallQual,OverallCond,BsmtFinSF1,TotalBsmtSF,GrLivArea,GarageCars,C
entralAir,GarageFinish,SaleCondition)

#Penalized Regression Methods: Ridge, Lasso, EN

#Check VIF scores to see which variables have inflated standard errors
vif(final.lm)
res = cor(final.train) #correlation between variables

#Ridge Regression (note that cv.glmnet by default does 10-fold cross-validation)
rr.glmnet = cv.glmnet(as.matrix(final.train[,1:10]), final.train$SalePrice, alpha=0)
attributes(rr.glmnet)
best.lambda <- rr.glmnet$lambda.min
ridge.coef = coef(rr.glmnet, s=best.lambda)
#RMSE
sqrt(rr.glmnet$cvm[rr.glmnet$lambda == rr.glmnet$lambda.1se])
#Predict
ridge.pred = predict(rr.glmnet, as.matrix(final.test), s=best.lambda)
```

```
#Lasso
lasso.glmnet = cv.glmnet(as.matrix(final.train[,1:10]), final.train$SalePrice, alpha=1)
attributes(lasso.glmnet)
best.lambda2 = lasso.glmnet$lambda.min
lasso.coef = coef(lasso.glmnet, s=best.lambda2)
lasso.coef[lasso.coef !=0] #47 variables
#RMSE
sqrt(lasso.glmnet$cvm[lasso.glmnet$lambda == lasso.glmnet$lambda.1se])
#Predict
lasso.pred = predict(lasso.glmnet, newx=as.matrix(final.test), s=best.lambda2)

#Elastic Net
tcontrol = trainControl(method="repeatedcv", number=10, repeats=5)

en.glmnet = train(as.matrix(final.train[,1:10]), final.train$SalePrice, trControl=tcontrol,
            method="glmnet", tuneLength=10)
attributes(en.glmnet)
en.glmnet$results
en.glmnet$bestTune
en.glmnet2 = en.glmnet$finalModel
en.coef = coef(en.glmnet2, s=en.glmnet$bestTune$lambda)
#RMSE
min(en.glmnet$results$RMSE)
#Predict
en.pred = predict(en.glmnet, as.matrix(final.test), s=en.glmnet$bestTune$lambda)

#Diagnostic plots for final model
par(mfrow=c(2,2))
plot(final.lm)

#Histogram of normal data showing the distribution of the studentized residuals
sresid <- rstudent(final.lm)
hist(sresid, freq=FALSE, main="Distribution of Studentized Residuals")
box()
xfit <- seq(min(sresid), max(sresid), length=40)
yfit <- dnorm(xfit)
lines(xfit, yfit, col='blue')

### Elastic Net has best RMSE but Lasso will get best kaggle score ###

#Write as Excel files

#Backwards Predictions
```

```
WriteXLS(data.frame(back.pred))
#Forward Predictions
WriteXLS(data.frame(forward.pred))
#Stepwise Predictions
WriteXLS(data.frame(step.pred))

#Ridge Regression Predictions
WriteXLS(data.frame(ridge.pred))
#Lasso Predictions
WriteXLS(data.frame(lasso.pred))
#EN Predictions
WriteXLS(data.frame(en.pred))

#Note- we converted the log values that we obtained (in excel) and submitted that for our
Kaggle scores
```