

House Case Study Report

Yimei Chen, Chris Dong, Qian Li, Jing Song

October 6, 2017

Loading the data and any packages

```
options("max.print"=10)
suppressMessages(library(tidyverse))
suppressMessages(library(magrittr))
suppressMessages(library(leaps))
suppressMessages(library(VIM))
suppressMessages(library(car))
suppressMessages(library(Hmisc))
suppressMessages(library(glmnet))
suppressMessages(library(grid))
suppressMessages(library(gridExtra))
suppressMessages(library(ggcorrplot))
suppressMessages(library(olsrr))
house <- read_csv("housing.txt", col_types = cols())
names(house) <- tolower(names(house))
house0 <- house
```

Exploratory Data Analysis

```
p1 <- house %>% ggplot(aes(x=grlivarea, y = saleprice,
                          color = factor(kitchenqual))) + geom_point(alpha = 0.5) +
  xlab("Above grade (ground) living area square feet") +
  ylab("Price of the house") + scale_y_continuous(label=scales::comma) +
  labs(colour = "Kitchen Quality") +
  theme(legend.title = element_text(size = 10, face = "bold"))
p2 <- house %>% ggplot(aes(x=log(grlivarea), y = log(saleprice),
                          color = factor(kitchenqual))) + geom_point(alpha = 0.5) +
  xlab("Log of Above grade (ground) living area square feet") +
  ylab("Log of Price of the house") + scale_y_continuous(label=scales::comma) +
  labs(colour = "Kitchen Quality") +
  theme(legend.title = element_text(size = 10, face = "bold"))

grid.arrange(p1,p2,ncol=1)
```

```
ggplot(house, aes(x=neighborhood,y=saleprice,color = factor(garagecars)))+geom_point(alpha = .5)+ theme
  ylab("Price of the house") + scale_y_continuous(label=scales::comma) +
  labs(colour = "Garage Car Capacity") +
  theme(legend.title = element_text(size = 10, face = "bold"))
```

```
house0%>% ggplot(aes(x=yearbuilt, y = saleprice,
                     color = factor(roofstyle))) + geom_point(alpha = 0.5) +
  xlab("Built Year") +
  ylab("Price of the house") + scale_y_continuous(label=scales::comma) +
  labs(colour = "Type of Roof") +
  theme(legend.title = element_text(size = 10, face = "bold"))
```

Convert mssubclass to factor and check for NAs

```
house$mssubclass <- factor(house$mssubclass)
house %>% sapply(function(x) sum(is.na(x))) %>% sort(decreasing = T)
```

```
##      poolqc  miscfeature      alley      fence  fireplacequ
##      1453      1406      1369      1179      690
## lotfrontage  garagetype  garageyrblt garagefinish  garagequal
##      259      81      81      81      81
## [ reached getOption("max.print") -- omitted 71 entries ]
```

Convert numeric variables that have NA to 0. Change `garageyrblt` to indicate whether or not the garage was built AFTER the house was built.

```
house$masvnrarea[which(is.na(house$masvnrarea))] <- 0
house$bsmtfintype1[which(is.na(house$bsmtfintype1))] <- 0
house$bsmtfintype2[which(is.na(house$bsmtfintype2))] <- 0
house$garageyrblt <- (house$garageyrblt > house$yearbuilt) * 1
house$garageyrblt[is.na(house$garageyrblt)] <- 0
```

Impute the NA in `lotfrontage`, `electrical` with K-Nearest Neighbors

```
k = round(sqrt(1460*.8) / 2)

house$lotfrontage <- kNN(house, variable = "lotfrontage", k = k)$lotfrontage
house$electrical <- kNN(house, variable = "electrical", k = k)$electrical
```

Convert all other NAs to "None"

```
house[is.na(house)] <- "None"
```

Make a new variable, `remodel` that indicates whether or not remodeling took place. Remove the `yearremodadd` variable because it is no longer needed. Make a new variable `soldminusbuilt` that indicates the number of years that it took for the house to get sold after getting built.

```
house$remodel <- T
house[house$yearbuilt == house$yearremodadd,]$remodel <- F
house$remodel <- as.numeric(house$remodel)
house %<>% select(-yearremodadd)

house$soldminusbuilt <- (house$yrsold - house$yearbuilt)
house %<>% select(-yrsold,-yearbuilt)
```

Combine all of the porch variables into one. Remove `id` because it is obviously not important.

```
house$porcharea <- with(house, openporchsf + enclosedporch +
  `3ssnporch` + screenporch)
house %<>% select(-id)
```

Change `lotshape` to a boolean whether or not it is Regular.

```
table(house$lotshape)
```

```
##
## IR1 IR2 IR3 Reg
## 484 41 10 925
```

```
house$lotshape <- (house$lotshape == 'Reg') * 1
```

Looking at the histogram of `mosold` we see many more houses being sold near summer time (and part of spring too) so we create a boolean. Most of the time, when we are creating a boolean, it is because it is insignificant otherwise.

```
house %>% ggplot(aes(x=mosold)) + geom_histogram(binwidth = 1) + xlim(0,13)+
  xlab("Month Sold") +
  ylab("Frequency")

house$summertime <- (house$mosold %in% 5:7) * 1
```

The next part of the code was very time-consuming but here's the general outline: It is similar to backwards selection but by hand and possibly more thorough because of the refactoring involved rather than simply removing it.

1. Check the p-value and significance for a particular variable.
2. If the variable is numeric and significant, keep it. If the variable is categorical and all levels are significant, keep it. If only some levels are significant then try to bin the factors into smaller number of levels to try and make them statistically significant. If nothing can be done, then remove the variable.
3. Repeat the above steps for the rest of the variables. Each time we remove a variable, we re-run the lm model to check if the Adjusted R Squared changed significantly or not.
4. When we finish going through all the variables, there will be about 30 ones left to consider.

```
house %<>% select(-mosold, -landcontour, -alley, -lotshape)

house$lotconfig <- (house$lotconfig == "Inside") * 1
house %<>% select(-lotconfig)
```

Here, we noticed lotfrontage became significant when we take the square root. We remove 1stflrsf, 2ndflrsf, lowqualfinsf because they make up the variable grlivarea. At first, we tried having all three of them and deleting grlivarea however we found that having just grlivarea performed better. We are deleting the porch variables because we have already aggregated them into porcharea.

```
fullmodel <- lm(saleprice~sqrt(lotfrontage)+porcharea+.,data = house)
summary(fullmodel)$r.squared
```

```
## [1] 0.9322597
```

```
house$condition1 <- relevel(factor(house$condition1), ref = "Norm")
house$condition2 <- relevel(factor(house$condition2), ref = "Norm")
```

```
house %<>% select(-roofstyle)
house %<>% select(-exterior2nd)
```

```
table(house$bldgtype)
```

```
##
## 1Fam 2fmCon Duplex Twnhs TwnhsE
## 1220 31 52 43 114
```

```
house <- house %>% select(-`1stflrsf`, -`2ndflrsf`, -lowqualfinsf,
  -totalbsmtsf, -openporchsf, -enclosedporch, -`3ssnporch`,
  -screenporch, -garagearea)
```

```
table(house$salecondition)
```

```
##
## Abnorml AdjLand Alloca Family Normal Partial
## 101 4 12 20 1198 125
```

```
house$salecondition <- (house$salecondition == "Normal") * 1
```

```
table(house$saletype)
```

```
##
##   COD   Con ConLD ConLI ConLw   CWD   New   Oth   WD
##   43    2    9    5    5    4   122    3  1267

house$saletype <- (house$saletype == 'New') * 1
house <- house %>% select(-saletype)

house$miscfeature <- (house$miscfeature != 'None') * 1
house %<>% select(-miscval, -miscfeature)

house$paveddrive <- (house$paveddrive == 'Y') * 1
house %<>% select(-paveddrive)

house$poolqc <- (house$poolqc != "None") * 1
house$fence <- (house$fence != "None") * 1
```

Here, I am changing the ordered factor into numeric. I want to make a correlation plot with every significant variable so I am converting all variables (as long as it makes sense) to numeric.

```
house$garagecond <- as.numeric(factor(house$garagecond,
  levels = c("None", "Po", "Fa", "TA", "Gd", "Ex"), labels = 0:5))
house$garagequal <- as.numeric(factor(house$garagequal,
  levels = c("None", "Po", "Fa", "TA", "Gd", "Ex"), labels = 0:5))

house %<>% select(-fence, -poolqc, -garagecond)

house %>% group_by(garagefinish) %>%
  summarise(avgprc = median(saleprice)) %>% arrange(desc(avgprc)) %>% head(2)

## # A tibble: 2 x 2
##   garagefinish avgprc
##   <chr>      <dbl>
## 1      Fin  215000
## 2      RFn  190000

house$garagefinish <- (house$garagefinish == "Fin") * 1
house %<>% select(-garagefinish)
```

Here, fireplacequ and fireplaces are obviously correlated so I choose the one that seems to explain saleprice better. However, they both end up being insignificant.

```
house$fireplacequ <- as.numeric(factor(house$fireplacequ,
  levels = c("None", "Po", "Fa", "TA", "Gd", "Ex"), labels = 0:5))
cor(house$saleprice, house$fireplacequ); cor(house$saleprice, house$fireplaces)

## [1] 0.5204376
## [1] 0.4669288

house %<>% select(-fireplacequ, -fireplaces)

house %<>% select(-garageyrblt)
house$garagetype <- relevel(factor(house$garagetype), ref = "None")

house$functional <- (house$functional == "Typ") * 1

house$kitchenqual <- as.numeric(factor(house$kitchenqual,
  levels = c("Po", "Fa", "TA", "Gd", "Ex"), labels = 1:5))
```

Similarly, totrmsabvgrd is highly correlated with grlivarea so I keep the better of the two.

```
cor(house$totrmsabvgrd ,house$saleprice);cor(house$grlivarea ,house$saleprice)
```

```
## [1] 0.5337232
```

```
## [1] 0.7086245
```

```
house %<>% select(-totrmsabvgrd)
```

I try to combine all of the bath variables but they end up not being significant so I just remove them.

```
table(house$fullbath)
```

```
##
```

```
## 0 1 2 3
```

```
## 9 650 768 33
```

```
house$bath <- house$fullbath + house$halfbath + house$bsmtfullbath + house$bsmthalfbath
```

```
house %<>% select(-fullbath, -halfbath, -bsmthalfbath, -bsmtfullbath)
```

```
house %<>% select(-bath)
```

```
house %>% group_by(electrical) %>% summarise(avgprc = median(saleprice)) %>% arrange(desc(avgprc))
```

```
## # A tibble: 5 x 2
```

```
##   electrical avgprc
```

```
##   <chr> <dbl>
```

```
## 1 SBrkr 170000
```

```
## 2 FuseA 121250
```

```
## 3 FuseF 115000
```

```
## 4 FuseP 82000
```

```
## 5 Mix 67000
```

```
house$electrical <- (house$electrical == "SBrkr") * 1
```

```
house %<>% select(-electrical, -centralair)
```

```
house$heatingqc <- as.numeric(factor(house$heatingqc,  
  levels = c("Po", "Fa", "TA", "Gd", "Ex"), labels = 1:5))
```

```
table(house$heatingqc)
```

```
##
```

```
## 1 2 3 4 5
```

```
## 1 49 428 241 741
```

```
house$heatingqc <- (house$heatingqc == 5) * 1
```

```
house %<>% select(-heating)
```

```
table(house$bsmtfintype1)
```

```
##
```

```
## 0 ALQ BLQ GLQ LwQ Rec Unf
```

```
## 37 220 148 418 74 133 430
```

```
house$bsmtfintype1 <- as.numeric(factor(house$bsmtfintype1,  
  levels = c("0", "Unf", "LwQ", "Rec", "BLQ", "ALQ", "GLQ"),  
  labels = 0:6))
```

```
house$bsmtfintype2 <- as.numeric(factor(house$bsmtfintype2,  
  levels = c("0", "Unf", "LwQ", "Rec", "BLQ", "ALQ", "GLQ"),  
  labels = 0:6))
```

```

house$bsmtfintype1 <- house$bsmtfintype1 + house$bsmtfintype2
house %<>% select(-bsmtfintype1, -bsmtfintype2)

house$bsmtexposure <- relevel(factor(house$bsmtexposure), ref = "None")

table(house$bsmtexposure)

##
##   None    Av    Gd    Mn    No
##    38   221   134   114   953

house %>% group_by(bsmtexposure) %>% summarise(avgprc = median(saleprice)) %>% arrange(desc(avgprc))

## # A tibble: 5 x 2
##   bsmtexposure avgprc
##         <fctr>   <dbl>
## 1             Gd 226975
## 2             Av 185850
## 3             Mn 182450
## 4             No 154000
## 5            None 104025

house$bsmtexposure <- (house$bsmtexposure == "Gd") * 1

house %>% group_by(bsmtcond) %>% summarise(avgprc = median(saleprice)) %>% arrange(desc(avgprc))

## # A tibble: 5 x 2
##   bsmtcond avgprc
##       <chr>   <dbl>
## 1         Gd 193879
## 2         TA 165000
## 3         Fa 118500
## 4       None 101800
## 5         Po  64000

table(house$bsmtcond)

##
##   Fa    Gd None    Po    TA
##   45   65   37     2 1311

house$bsmtcond <- as.numeric(factor(house$bsmtcond,
  levels = c("None", "Po", "Fa", "TA", "Gd", "Ex"),
  labels = 0:5))

house$bsmtqual <- as.numeric(factor(house$bsmtqual,
  levels = c("None", "Po", "Fa", "TA", "Gd", "Ex"),
  labels = 0:5))
cor(house$bsmtcond, house$bsmtqual)

## [1] 0.6337134

cor(house$bsmtcond, house$saleprice); cor(house$bsmtqual, house$saleprice)

## [1] 0.2126072
## [1] 0.5852072

```

```

house %<>% select(-bsmtcond)
house %<>% select(-bsmtqual)

table(house$foundation)

##
## BrkTil CBlock PConc Slab Stone Wood
## 146 634 647 24 6 3

house %>% group_by(foundation) %>% summarise(avgprc = median(saleprice)) %>% arrange(desc(avgprc))

## # A tibble: 6 x 2
## foundation avgprc
## <chr> <dbl>
## 1 PConc 205000
## 2 Wood 164000
## 3 CBlock 141500
## 4 Stone 126500
## 5 BrkTil 125250
## 6 Slab 104150

house$foundation <- (house$foundation == "PConc")*1

house$extercond <- as.numeric(factor(house$extercond,
  levels = c("Po", "Fa", "TA", "Gd", "Ex"),
  labels = 1:5))
house$exterqual <- as.numeric(factor(house$exterqual,
  levels = c("Po", "Fa", "TA", "Gd", "Ex"),
  labels = 1:5))
cor(house$extercond, house$exterqual)

## [1] 0.00918398

house$masvnrtype <- relevel(factor(house$masvnrtype), ref = "None")

table(house$masvnrtype)

##
## None BrkCmn BrkFace Stone
## 872 15 445 128

house$masvnrtype <- (house$masvnrtype != "None") * 1

Boolean whether or not housestyle is either 2Story or 2.5Fin.

table(house$housestyle)

##
## 1.5Fin 1.5Unf 1Story 2.5Fin 2.5Unf 2Story SFoyer SLvl
## 154 14 726 8 11 445 37 65

house %>% group_by(housestyle) %>% summarise(avgprc = median(saleprice)) %>% arrange(desc(avgprc))

## # A tibble: 8 x 2
## housestyle avgprc
## <chr> <dbl>
## 1 2.5Fin 194000
## 2 2Story 190000

```

```

## 3      SLvl 164500
## 4      1Story 154750
## 5      SFoyer 135960
## 6      2.5Unf 133900
## 7      1.5Fin 132000
## 8      1.5Unf 111250

house$housestyle <- (house$housestyle == "2Story" |
                     house$housestyle == "2.5Fin")*1

table(house$bldgtype)

##
##      1Fam 2fmCon Duplex  Twnhs TwnhsE
##      1220    31    52    43    114

house$bldgtype <- (house$bldgtype == "1Fam" | house$bldgtype == "2FmCon") * 1
house %<>% select(-bldgtype)

table(house$landslope)

##
##      Gtl  Mod  Sev
##      1382   65   13

house$landslope <- (house$landslope == "Gtl") * 1
house %<>% select(-landslope)

table(house$utilities)

##
##      AllPub NoSeWa
##      1459      1

house %<>% select(-utilities, -street)

house %>% group_by(mszoning) %>% summarise(avgprc = median(saleprice)) %>% arrange(desc(avgprc))

## # A tibble: 5 x 2
##   mszoning avgprc
##   <chr>   <dbl>
## 1      FV 205950
## 2      RL 174000
## 3      RH 136500
## 4      RM 120500
## 5 C (all)  74700

table(house$mszoning)

##
## C (all)      FV      RH      RL      RM
##      10      65      16    1151    218

house$mszoning <- relevel(factor(house$mszoning), ref = "RL")

house %<>% select(-mszoning)

house %>% group_by(mssubclass) %>% summarise(avgprc = median(saleprice)) %>% arrange(desc(avgprc))

```



```
## # A tibble: 15 x 2
##   mssubclass avgprc
##   <fctr>   <dbl>
## 1         60 215200
## 2        120 192000
## 3         80 166500
## 4         75 163500
## 5         20 159250
## 6         70 156000
## 7        160 146000
## 8         40 142500
## 9         85 140750
## 10        90 135980
## 11         50 132000
## 12        190 128250
## 13         45 107500
## 14         30  99900
## 15        180  88500

house %<>% select(-mssubclass, -lotfrontage, -porcharea, -extercond, -foundation,
                 -exterior1st)

house %>% group_by(condition1) %>% summarise(avgprc = median(saleprice)) %>% arrange(desc(avgprc))

## # A tibble: 9 x 2
##   condition1 avgprc
##   <fctr>   <dbl>
## 1      RRNn 214000
## 2      PosA 212500
## 3      PosN 200000
## 4      RRNe 190750
## 5      RRAn 171495
## 6      Norm 166500
## 7      RRAe 142500
## 8      Feedr 140000
## 9      Artery 119550

house$condition1 <- (house$condition1 == "Artery" | house$condition1 == "Feedr" |
  house$condition1 == "RRAe")*1
house$condition2 <- (house$condition2 == "PosN") * 1

cor(house$garageequal, house$garagecars)

## [1] 0.5766224

house %<>% select(-garageequal)

fullmodel <- lm(saleprice~., data = house)
summary(fullmodel)

##
## Call:
## lm(formula = saleprice ~ ., data = house)
##
## Residuals:
```

```
##      Min      1Q  Median      3Q      Max
## -189220 -12119      844   12154  189220
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -7.000e+05  3.478e+04 -20.126  < 2e-16 ***
## lotarea         4.807e-01  8.390e-02   5.730  1.23e-08 ***
## [ reached getOption("max.print") -- omitted 61 rows ]
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26740 on 1397 degrees of freedom
## Multiple R-squared:  0.8915, Adjusted R-squared:  0.8867
## F-statistic: 185.2 on 62 and 1397 DF,  p-value: < 2.2e-16
```

Checking multicollinearity. Looks good. For the generalized variance inflation factor (normalized by the degree of freedom), everything except one is less than 2.

```
vif(fullmodel)
```

```
##              GVIF Df GVIF^(1/(2*Df))
## lotarea         1.431345  1         1.196389
## neighborhood    78.751984 24         1.095230
## condition1       1.183865  1         1.088056
## [ reached getOption("max.print") -- omitted 25 rows ]
```

Interestingly, soldminusbuilt which is yrsold - yearbuilt becomes insignificant in this smaller model with only the best predictors

```
house_numeric <- house[,apply(house,function(x) is.numeric(x))]
house_numeric %<>% select(saleprice, everything())
bestpredictors <- names(house_numeric)[apply(house_numeric,
function(x) abs(cor(house_numeric$saleprice, x))) >= 0.5][-1]
```

```
bestpredictors <- bestpredictors[-6]
```

```
bestmodel <- lm(saleprice~overallqual + exterqual + grlivarea +
  kitchenqual + garagecars + neighborhood, data = house)
```

```
summary(bestmodel)$r.squared
```

```
## [1] 0.808378
```

Subset with only best predictors

```
housesubset <- house %>% select(bestpredictors)
```

So, 6 variables capture 0.808378 of the variation in sale price for our model.

Checking assumptions.

```
cor(housesubset)
vif(bestmodel)
```

```
g1 <- ggplot(housesubset, aes(sample = grlivarea)) + stat_qq() + ggtitle("grlivarea")
g2 <- ggplot(housesubset, aes(sample = log(grlivarea))) + stat_qq() + ggtitle("log(grlivarea)")
g3 <- ggplot(house, aes(sample = saleprice)) + stat_qq() + ggtitle("saleprice")
```

```
g4 <- ggplot(house, aes(sample = log(saleprice))) + stat_qq() + ggtitle("log(saleprice)")
  grid.arrange(g1,g2,g3,g4)
```

```
bestmodel2 <- lm(log(saleprice)~overallqual + exterqual + log(grlivarea) +
  kitchenqual + garagecars + neighborhood, data = house)
summary(bestmodel2)
```

```
##
## Call:
## lm(formula = log(saleprice) ~ overallqual + exterqual + log(grlivarea) +
##     kitchenqual + garagecars + neighborhood, data = house)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97098 -0.07887  0.01184  0.09490  0.52805
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      8.200823    0.119367  68.703 < 2e-16 ***
## overallqual       0.085902    0.005543  15.497 < 2e-16 ***
## [ reached getOption("max.print") -- omitted 28 rows ]
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1567 on 1430 degrees of freedom
## Multiple R-squared:  0.8492, Adjusted R-squared:  0.8462
## F-statistic: 277.7 on 29 and 1430 DF,  p-value: < 2.2e-16
```

exterqual becomes insignificant once we take the log of the response variable

```
bestmodel3 <- lm(log(saleprice)~overallqual + log(grlivarea) +
  kitchenqual + garagecars + neighborhood, data = house)
summary(bestmodel3)$r.squared
```

```
## [1] 0.8488445
```

Check for high leverage points. There are 98 high leverage points.

```
( high_leverage <- as.numeric(names(hatvalues(bestmodel3))[(hatvalues(bestmodel3) > 2*ncol(house)/nrow(house))])
```

```
lev_df <- data_frame(rstudent = rstudent(bestmodel3),
  hatvalue = hatvalues(bestmodel3))
lev_df$highlev <- F
lev_df[high_leverage,]$highlev <- T
lev_df %>% ggplot(aes(x=hatvalue, y = rstudent,color = highlev)) + geom_point()+
  xlab("Hat Values") +
  ylab("Sstandardized Residuals") + scale_y_continuous(label=scales::comma) +
  labs(colour = "High Leverage?") +
  theme(legend.title = element_text(size = 10, face = "bold"))
```

```
length(hatvalues(bestmodel3)[(hatvalues(bestmodel3) > 2*ncol(house)/nrow(house))])
```

```
hatvalues(bestmodel)[hatvalues(bestmodel3) > 0.5]
```

```
infm <- influence.measures(bestmodel3)
threshold <- sqrt(2*ncol(house)/nrow(house))
```

Check for influence points. There are 184 high influence points with a threshold of $\sqrt{\frac{p}{n}} = 0.1993139$

```
(high_influence <- which(abs(infm$infmt[,30])>threshold))

## 4 5 18 24 31 53 54 57 103 105
## 4 5 18 24 31 53 54 57 103 105
## [ reached getOption("max.print") -- omitted 186 entries ]

inf_df <- data_frame(dffits = dffits(bestmodel3), index = 1:nrow(house))
inf_df$highinf <- F
inf_df[high_influence,]$highinf <- T

inf_df %>% ggplot(aes(x=index, y=dffits, color = highinf)) + geom_point() +
  xlab("Observation Number") +
  ylab("DFFITS") + scale_y_continuous(label=scales::comma) +
  labs(colour = "High Influence Point?") +
  theme(legend.title = element_text(size = 10, face = "bold"))

influence <- ols_dffits_plot(bestmodel3)
```

Let's examine Observation # 1299, and 524

```
house[1299,] %>% View()
house[542,] %>% View()

bestmodel4 <- lm(log(saleprice)~overallqual + log(grlivarea) +
  kitchenqual + garagecars + neighborhood, data = house[c(-1299,-542),])
summary(bestmodel4)$r.squared
```

```
## [1] 0.8530995
```

By just removing two points, our Adjusted R-squared went from 0.8458869 to 0.8502211

There are 89 outliers. Let's see what happens if we simply remove the outliers.

```
influenceindex <- unlist(influence$outliers[1])

bestmodelnoinfluence <- lm(log(saleprice)~overallqual + log(grlivarea) +
  kitchenqual + garagecars + neighborhood, data = house[-influenceindex,])
summary(bestmodelnoinfluence)$r.squared
```

```
## [1] 0.8889236
```

We see that our Adjusted R-squared went from 0.8502211 to 0.8866905 after removing ALL the influence points.

```
t1 <- names(house)[1:11]
t2 <- names(house)[12:21]
t2[11] <- ""
t3 <- names(house)[22:31]
t3[11] <- ""

data_frame(t1,t2,t3) %>%
  knitr::kable(col.names = c("", "", ""))
```

lotarea	bsmtexposure	garagetype
neighborhood	bsmtfinsf1	garagecars
condition1	bsmtfinsf2	wooddecksf
condition2	bsmtunfsf	poolarea

housestyle	heatingqc	salecondition
overallqual	grlivarea	saleprice
overallcond	bedroomabvgr	remodel
roofmatl	kitchenabvgr	soldminusbuilt
masvnrtype	kitchenqual	NA
masvnrarea	functional	NA
exterqual		

```
house2 <- house
house2[influenceindex, ]$saleprice <- NA
house2$saleprice <- kNN(house2, variable = "saleprice", k = k)$saleprice
```

```
## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx,
## numericalX, : NAs introduced by coercion
```

```
## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx,
## numericalX, : NAs introduced by coercion
```

```
## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx,
## numericalX, : NAs introduced by coercion
```

```
## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx,
## numericalX, : NAs introduced by coercion
```

```
bestmodelimputeinfluence <- lm(log(saleprice)~overallqual + log(grlivarea) +
  kitchenqual + garagecars + neighborhood, data = house2)
summary(bestmodelimputeinfluence)$r.squared
```

```
## [1] 0.8674347
```

Let's try our model with all of the relevant variables. First, we notice that the R squared improves by taking the log of saleprice, lotarea, grlivarea and the square root of bsmtfinsf1. We also notice that housestyle and masvnrtype is no longer significant so we remove them.

```
model31var <- lm(log(saleprice) ~ log(lotarea) +
  sqrt(bsmtfinsf1)+log(grlivarea)+., data = house)
summary(model31var)$r.squared
```

```
## [1] 0.9211975
```

Accounting for outliers in the full model through imputation

```
model31varimpute <- lm(log(saleprice) ~ log(lotarea) +
  sqrt(bsmtfinsf1)+log(grlivarea)+., data = house2)
summary(model31varimpute)$r.squared
```

```
## [1] 0.9216878
```

We can try removing the outliers, which improved the R squared by a lot.

I remove some variables found to be insignificant.

```
house3 <- house2 %>% select(-condition2,-roofmatl,-garagetype,-poolarea,-remodel)
```

Remove exterqual

```
house4 <- house3 %>% select(-exterqual)
```

Getting all of the numeric variables.

```
house_numeric <- house4[,sapply(house4,function(x) is.numeric(x))]

house_numeric %<>% select(saleprice, everything())
#install.packages("ggcorrplot")

cor_matrix <- cor(house_numeric)

ggcorrplot(cor_matrix, type = "lower", outline.col = "white", insig = "blank")
```

FINAL MODEL

I test the multicollinearity, significance of variables in the model, normality for our final model.

```
endmodel <- lm(log(saleprice) ~ log(lotarea) +
               sqrt(bsmtfinsf1)+log(grlivarea) + . -
               lotarea - bsmtfinsf1 - grlivarea,
               data = house4[-influenceindex,])
vifmodel <- lm(log(saleprice) ~ log(lotarea) +
               sqrt(bsmtfinsf1)+log(grlivarea) + . -
               lotarea - bsmtfinsf1 - grlivarea - neighborhood,
               data = house4[-influenceindex,])
vif(vifmodel) %>% knitr::kable()
```

log(lotarea)	1.422712
sqrt(bsmtfinsf1)	2.781591
log(grlivarea)	4.183011
condition1	1.101296
housestyle	2.172729
overallqual	3.501096
overallcond	1.411713
masvnrtype	2.193630
masvnrarea	2.111939
bsmtexposure	1.144368
bsmtfinsf2	1.167915
bsmtunfsf	3.089384
heatingqc	1.493263
bedroomabvgr	1.784709
kitchenabvgr	1.196703
kitchenqual	2.223142
functional	1.164553
garagecars	1.997033
wooddecksf	1.141370
salecondition	1.087261
soldminusbuilt	2.820567

```
options(max.print=999)
summary(endmodel)
```

```
##
## Call:
## lm(formula = log(saleprice) ~ log(lotarea) + sqrt(bsmtfinsf1) +
```

```
##      log(grlivarea) + . - lotarea - bsmtfinsf1 - grlivarea, data = house4[-influenceindex,
##      ])
##
## Residuals:
##      Min        1Q      Median        3Q        Max
## -0.40974 -0.05061  0.00433  0.05282  0.30491
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.997e+00  1.119e-01  62.524 < 2e-16 ***
## log(lotarea)    1.015e-01  7.935e-03  12.787 < 2e-16 ***
## sqrt(bsmtfinsf1) 4.891e-03  3.327e-04  14.703 < 2e-16 ***
## log(grlivarea)  4.563e-01  1.625e-02  28.082 < 2e-16 ***
## neighborhoodBrDale -8.810e-02  3.680e-02  -2.394 0.016799 *
## neighborhoodBrkSide -1.172e-02  3.078e-02  -0.381 0.703513
## neighborhoodClearCr  1.093e-02  3.450e-02   0.317 0.751567
## neighborhoodCollgCr -2.114e-02  2.693e-02  -0.785 0.432537
## neighborhoodCrawfor  1.003e-01  3.128e-02   3.208 0.001369 **
## neighborhoodEdwards -8.649e-02  2.908e-02  -2.974 0.002990 **
## neighborhoodGilbert -2.384e-02  2.861e-02  -0.833 0.404723
## neighborhoodIDOTRR -1.029e-01  3.426e-02  -3.004 0.002716 **
## neighborhoodMeadowV -8.378e-02  3.583e-02  -2.338 0.019518 *
## neighborhoodMitchel -5.248e-02  3.001e-02  -1.749 0.080592 .
## neighborhoodNAMES  -4.581e-02  2.805e-02  -1.633 0.102667
## neighborhoodNoRidge  4.277e-02  3.121e-02   1.370 0.170778
## neighborhoodNPkVill -1.141e-02  4.029e-02  -0.283 0.776982
## neighborhoodNridgHt  8.502e-02  2.807e-02   3.029 0.002503 **
## neighborhoodNWAmes  -5.526e-02  2.912e-02  -1.898 0.057936 .
## neighborhoodOldTown -8.858e-02  3.010e-02  -2.943 0.003302 **
## neighborhoodSawyer  -1.865e-02  2.969e-02  -0.628 0.529952
## neighborhoodSawyerW -4.491e-02  2.892e-02  -1.553 0.120675
## neighborhoodSomerst  5.809e-02  2.723e-02   2.134 0.033049 *
## neighborhoodStoneBr  1.187e-01  3.429e-02   3.461 0.000556 ***
## neighborhoodSWISU  -5.297e-02  3.523e-02  -1.504 0.132891
## neighborhoodTimber  -9.012e-03  3.053e-02  -0.295 0.767936
## neighborhoodVeenker  3.633e-03  4.258e-02   0.085 0.932023
## condition1      -6.183e-02  8.942e-03  -6.915 7.27e-12 ***
## housestyle      -1.961e-02  8.231e-03  -2.382 0.017341 *
## overallqual      5.486e-02  3.605e-03  15.217 < 2e-16 ***
## overallcond      3.908e-02  2.795e-03  13.983 < 2e-16 ***
## masvnrtype      -1.712e-02  7.803e-03  -2.195 0.028360 *
## masvnrarea       5.736e-05  2.178e-05   2.633 0.008553 **
## bsmtexposure     4.713e-02  9.930e-03   4.747 2.29e-06 ***
## bsmtfinsf2       7.649e-05  1.707e-05   4.480 8.11e-06 ***
## bsmtunfsf        7.052e-05  1.002e-05   7.039 3.09e-12 ***
## heatingqc        2.391e-02  6.380e-03   3.748 0.000186 ***
## bedroomabvgr     -1.348e-02  4.343e-03  -3.105 0.001945 **
## kitchenabvgr     -5.762e-02  1.318e-02  -4.373 1.32e-05 ***
## kitchenqual      4.249e-02  5.804e-03   7.321 4.26e-13 ***
## functional       7.591e-02  1.112e-02   6.829 1.30e-11 ***
## garagecars       5.012e-02  4.897e-03  10.234 < 2e-16 ***
## wooddecksf       7.271e-05  2.201e-05   3.304 0.000980 ***
## salecondition     3.020e-03  6.998e-03   0.432 0.666128
## soldminusbuilt   -2.284e-03  2.145e-04 -10.644 < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09123 on 1326 degrees of freedom
## Multiple R-squared:  0.9406, Adjusted R-squared:  0.9386
## F-statistic: 477.1 on 44 and 1326 DF,  p-value: < 2.2e-16
ks.test(endmodel$residuals, pnorm, mean(endmodel$residuals),
        sd(endmodel$residuals))

##
## One-sample Kolmogorov-Smirnov test
##
## data:  endmodel$residuals
## D = 0.040517, p-value = 0.02219
## alternative hypothesis: two-sided
ncvTest(endmodel)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 3.569303    Df = 1    p = 0.05885702
resid_df <- data_frame(res = endmodel$residuals)

r1 <- ggplot(endmodel, aes(.fitted, .resid)) + geom_point() + xlab("Fitted Values") + ylab("Residuals") +
  ggtitle("Residuals vs Fitted Values")

r2 <- ggplot(endmodel, aes(qqnorm(.stdresid)[[1]], .stdresid)) + geom_point(na.rm = T) + geom_abline(int,
  ylab("Standard Residuals") + ggtitle("QQ Residual Plot")

grid.arrange(r1,r2,ncol=2)
```

Checking with LASSO if any variables to remove. Although LASSO recommends to delete `bsmtunsf` and `bedroomabvgr`, removing them lowers the R squared so I will keep them. Many of the neighborhoods are in fact significant so I will leave the non-significant levels in the model anyway.

```
lassorefactor <- function(){
  x <- model.matrix(saleprice ~ ., data = house4)[,-1]
  y <- house$saleprice
  train <- sample(1:nrow(x), nrow(x) / 2)
  test <- (-train)
  y.train <- y[train]
  y.test <- y[test]
  grid.lambda <- 10^seq(10, -2, length = 100)
  lasso.model <- glmnet(x, y, alpha = 1, lambda = grid.lambda)
  set.seed(1)
  cv.out <- cv.glmnet(x[train,], y.train, alpha = 1)
  best.lambda <- cv.out$lambda.min
  lasso.pred <- predict(lasso.model, s = best.lambda, newx = x[test,])
  mspe.lasso <- mean((lasso.pred - y.test)^2)
  final.model <- glmnet(x, y, alpha = 1, lambda = best.lambda)
  c <- coef(final.model)
  ind <- which(c==0)
  variables <- row.names(c)[ind]
  return(variables)
```



```
}
```

```
lassorefactor()
```

```
## [1] "neighborhoodBlueste" "neighborhoodBrkSide" "neighborhoodClearCr"
## [4] "neighborhoodCollgCr" "neighborhoodGilbert" "neighborhoodIDOTRR"
## [7] "neighborhoodMeadowV" "neighborhoodMitchel" "neighborhoodNAMES"
## [10] "neighborhoodNPkVill" "neighborhoodNWAmes" "neighborhoodSawyer"
## [13] "neighborhoodSawyerW" "neighborhoodSomerset" "neighborhoodSWISU"
## [16] "neighborhoodTimber" "neighborhoodVeenker" "masvnrtype"
## [19] "bsmtfinsf2" "bsmtunfsf" "bedroomabvgr"
## [22] "salecondition"
```

Thus, our final model includes the following variables:

```
names(house4)
```

```
## [1] "lotarea" "neighborhood" "condition1" "housestyle"
## [5] "overallqual" "overallcond" "masvnrtype" "masvnrarea"
## [9] "bsmtexposure" "bsmtfinsf1" "bsmtfinsf2" "bsmtunfsf"
## [13] "heatingqc" "grlivarea" "bedroomabvgr" "kitchenabvgr"
## [17] "kitchenqual" "functional" "garagecars" "wooddecksf"
## [21] "salecondition" "saleprice" "soldminusbuilt"
```

```
signif_var <- house4 %>% select(-neighborhood) %>%
  sapply(function(x) abs(cor(x,house4$saleprice)))
```

```
signif_var[signif_var >= 0.5]
```

```
## overallqual grlivarea kitchenqual garagecars saleprice
## 0.8134559 0.7018887 0.6839772 0.6646509 1.0000000
## soldminusbuilt
## 0.5655127
```

```
summary(lm(log(saleprice)~log(grlivarea) +kitchenqual +garagecars + soldminusbuilt + overallqual, data = house4))
```

```
##
```

```
## Call:
```

```
## lm(formula = log(saleprice) ~ log(grlivarea) + kitchenqual +
## garagecars + soldminusbuilt + overallqual, data = house4)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.75832 -0.08904  0.00792  0.09445  0.53539
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.3153231  0.0984908  84.427  <2e-16 ***
## log(grlivarea) 0.3965268  0.0155380  25.520  <2e-16 ***
## kitchenqual    0.0799063  0.0081813   9.767  <2e-16 ***
## garagecars     0.0746529  0.0070858  10.536  <2e-16 ***
## soldminusbuilt -0.0023061  0.0001725 -13.368  <2e-16 ***
## overallqual    0.0823649  0.0047637  17.290  <2e-16 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Residual standard error: 0.1479 on 1454 degrees of freedom
```

```
## Multiple R-squared:  0.8389, Adjusted R-squared:  0.8383
## F-statistic:  1514 on 5 and 1454 DF,  p-value: < 2.2e-16
```

Part I: Explanatory Modeling

TASK 1

The five most relevant features that are most relevant in determining a house's sale price are `overallqual`, `grlivarea`, `kitchenqual`, `garagecars`, and `soldminusbuilt`. The fifth variable, `soldminusbuilt` is equal to `yearsold - yearbuilt`.

TASK 2

```
morty<- read_csv("Morty.txt", col_types = cols())
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

Function to transform TEST DATA accordingly. Please run the function `transform()` and provide the data frame to the argument

```
transform <- function(df){
  names(df) <- tolower(names(df))

  df[is.na(df)] <- "None"
  df$soldminusbuilt <- (df$yrsold - df$yearbuilt)
  df$summertime <- (df$mosold %in% 5:7) * 1
  df$saletype <- (df$saletype == 'New') * 1

  df %<>% select(intersect(names(df), names(house4)))

  df$condition1 <- (df$condition1 == "Artery" |
    df$condition1 == "Feedr" | df$condition1 == "RR Ae") * 1

  df$housestyle <- (df$housestyle == "2Story" |
    df$housestyle == "2.5Fin") * 1

  df$masvnrtype <- (df$masvnrtype != "None") * 1
  df$bsmtexposure <- (df$bsmtexposure == "Gd") * 1

  df$heatingqc <- as.numeric(factor(df$heatingqc,
    levels = c("Po", "Fa", "TA", "Gd", "Ex"), labels = 1:5))

  df$kitchenqual <- as.numeric(factor(df$kitchenqual,
    levels = c("Po", "Fa", "TA", "Gd", "Ex"), labels = 1:5))

  df$functional <- (df$functional == "Typ") * 1
  df$salecondition <- (df$salecondition == "Normal") * 1
  return(df)
```

```
}
morty2 <- transform(morty)
```

morty2 is our transformed data. Note that it only has 25 variables

```
confmorty <- exp(predict(endmodel, morty2, interval = "confidence", level = 0.95))
confmorty %>% knitr::kable()
```

	fit	lwr	upr
	186224.5	174489	198749.4

```
morty_stat <- as.numeric(unlist(morty2))
```

```
## Warning: NAs introduced by coercion
```

```
names(morty_stat) <- names(morty2)
mean_stat <- sapply(house4, function(x) round(mean(x)))
```

```
## Warning in mean.default(x): argument is not numeric or logical: returning
## NA
```

```
morty_stat
```

```
##      lotarea  neighborhood  condition1  housestyle  overallqual
##      14115      NA          0          0          5
##      overallcond  masvnrtype  masvnrarea  bsmtexposure  bsmtfinsf1
##      5          0          0          0          732
##      bsmtfinsf2  bsmtunfsf  heatingqc  grlivarea  bedroomabvgr
##      0          64          5          1362          1
##      kitchenabvgr  kitchenqual  functional  garagecars  wooddecksf
##      1          3          1          2          40
##      salecondition  saleprice  soldminusbuilt
##      1          143000          16
```

```
mean_stat
```

```
##      lotarea  neighborhood  condition1  housestyle  overallqual
##      10517      NA          0          0          6
##      overallcond  masvnrtype  masvnrarea  bsmtexposure  bsmtfinsf1
##      6          0          103          0          444
##      bsmtfinsf2  bsmtunfsf  heatingqc  grlivarea  bedroomabvgr
##      47          567          1          1515          3
##      kitchenabvgr  kitchenqual  functional  garagecars  wooddecksf
##      1          4          1          2          94
##      salecondition  saleprice  soldminusbuilt
##      1          179380          37
```

```
(improve <- house4 %>% select(-neighborhood, -saleprice, -soldminusbuilt) %>% sapply(function(x) abs(co
```

```
##      overallqual  grlivarea  kitchenqual  garagecars  masvnrarea
##      0.81345585  0.70188872  0.68397717  0.66465095  0.48383990
##      heatingqc  masvnrtype  bsmtfinsf1  wooddecksf  bsmtexposure
##      0.45175604  0.40262369  0.38668633  0.31654464  0.27827562
##      housestyle  bsmtunfsf  lotarea  condition1  bedroomabvgr
##      0.26080649  0.23979075  0.20994846  0.18742710  0.16286668
##      salecondition  kitchenabvgr  functional  overallcond  bsmtfinsf2
```

```
##      0.15899900      0.14185984      0.12634291      0.11069778      0.02201662
```

```
improve %>% knitr::kable()
```

overallqual	0.8134559
grlivarea	0.7018887
kitchenqual	0.6839772
garagecars	0.6646509
masvnrarea	0.4838399
heatingqc	0.4517560
masvnrtype	0.4026237
bsmtfinsf1	0.3866863
wooddecksf	0.3165446
bsmtexposure	0.2782756
housestyle	0.2608065
bsmtunfsf	0.2397908
lotarea	0.2099485
condition1	0.1874271
bedroomabvgr	0.1628667
salecondition	0.1589990
kitchenabvgr	0.1418598
functional	0.1263429
overallcond	0.1106978
bsmtfinsf2	0.0220166

`overallqual` and `kitchenqual` are in the top 3 for correlation with `saleprice`. `grlivarea` is difficult/nearly impossible to improve so we will move on to the next variable.

Conclusion: Morty should try to improve the `overallqual`, which is the overall material and finish of the house. This may mean repainting some areas on the house to make it look nicer. Morty currently has a rating of 5 out 10 (average rating is 6 out of 10) so there is definitely room for improvement. Next, Morty should improve `kitchenqual`, which is kitchen quality. Maybe, there can be some remodeling done or fixing anything that is either old, or possibly broken. Morty has a rating of 3 out of 5 compared to the average rating of 4 out of 5. Finally, he can increase `garagecars`. After doing some research, it is possible to extend a garage. Although we removed `garagearea` since it is correlated with `garagecars`, both have high correlation with `saleprice` so Morty can consider to extend his garage – it may be worth the investment.

We believe that Morty can sell his house for a *maximum* of 198,749.4. The 95 % confidence interval goes from 186,224.5 to 198,749.4 with an average of 174,489.

Part II Predictive Modeling

Ordinary Least Squares

```
set.seed(1)
train <- sample(nrow(house)*.8)
test <- (-train)
housetrain <- house4[train,]
housetest <- house4[test,]

OLS_train <- lm(log(saleprice) ~ log(lotarea) +
               sqrt(bsmtfinsf1)+log(grlivarea) + . -
               lotarea - bsmtfinsf1 - grlivarea,
```

```

        data = housetrain[-influenceindex,])
OLS_predict <- exp(predict(OLS_train, housetest,
        interval = "prediction", level = 0.95, type = "response"))
prettyNum(mean((OLS_predict[,1] - housetrain$saleprice)^2), big.mark = ",")

## [1] "10,678,431,344"

GLS_train <- glm(log(saleprice) ~ log(lotarea) +
        sqrt(bsmtfinsf1)+log(grlivarea) + . -
        lotarea - bsmtfinsf1 - grlivarea,
        data = housetrain[-influenceindex,])
GLS_predict <- exp(predict(OLS_train, housetest,
        interval = "prediction", level = 0.95, type = "response"))
prettyNum(mean((GLS_predict[,1] - housetrain$saleprice)^2), big.mark = ",")

## [1] "10,678,431,344"

```

Define the function to generate models for ridge, lasso and elastic net

```

model_func <- function(input_data, input_alpha){
  set.seed(1)
  x <- model.matrix(saleprice ~ ., data = input_data)[,-1]
  y <- house$saleprice
  train <- sample(nrow(house)*.8)
  test <- (-train)
  y.train <- y[train]
  y.test <- y[test]
  grid.lambda <- 10^seq(10, -2, length = 100)
  model.train <- glmnet(x[train, ], y.train, alpha = input_alpha, lambda = grid.lambda)
  set.seed(1)
  cv.out <- cv.glmnet(x[train,], y.train, alpha = input_alpha)
  best.lambda <- cv.out$lambda.min
  pred <- predict(model.train, s = best.lambda, newx = x[test,])
  mspe <- mean((pred - y.test)^2)
  final.model <- glmnet(x, y, alpha = input_alpha, lambda = best.lambda)
  c <- coef(final.model)
  return(c(mspe, final.model))
}

```

Ridge regression model, λ set at 0

```

ridge_result <- model_func(house4,0)
ridge_mspe <- ridge_result[1]
prettyNum(ridge_mspe, big.mark = ",")

```

```

##
## "1,712,490,366"

```

lasso regression model, lambda set at 1

```

lasso_result <- model_func(house4,1)
lasso_mspe <- lasso_result[1]
prettyNum(lasso_mspe, big.mark = ",")

```

```
##
## "1,821,002,807"
```

elastic net regression, lambda set at 0.5

```
elastic_result <- model_func(house4,0.5)
elastic_mspe <- elastic_result[1]
prettyNum(elastic_mspe, big.mark = ",")
```

```
##
## "1,815,619,193"
```

λ is chosen to determine whether we are performing Ridge ($\lambda = 0$), Lasso ($\lambda = 1$), Elastic Net ($\lambda = 0.5$). The tuning parameters in the respective models is chosen via cross validation after trying 100 different ones.

```
help(cv.glmnet)
```

Justification

Our ridge model performed the best and has the lowest MSPE.

```
countzero <- function(x){
  sum(x==0)
}
apply(house4, function(x) countzero(x))
```

```
##      lotarea  neighborhood  condition1  housestyle  overallqual
##      0        0            1320        1007          0
## overallcond  masvnrarea  masvnrtype  bsmtexposure  bsmtfinsf1
##      0        872        869        1326          467
## bsmtfinsf2   bsmtunfsf   heatingqc   grlivarea   bedroomabvgr
##      1293      118        719        0           6
## kitchenabvgr kitchenqual  functional  garagecars  wooddecksf
##      1        0          100        81          761
## salecondition  saleprice  soldminusbuilt
##      262        0          64
```

Many of these are boolean variables, but we can see that `masvnrarea`, `bsmtfinsf`, `bsmtfinsf2`, and `bsmtunfsf` all have zeros. We chose all of these variables because we found them to be statistically significant in our model.

```
house4 %>% select(-neighborhood) %>% apply(function(x) abs(cor(x, house4$saleprice))) %>% sort(decreas
```

```
##      saleprice  overallqual  grlivarea  kitchenqual  garagecars
##      1.00000000  0.81345585  0.70188872  0.68397717  0.66465095
## soldminusbuilt  masvnrarea  heatingqc  masvnrtype  bsmtfinsf1
##      0.56551269  0.48383990  0.45175604  0.40262369  0.38668633
## wooddecksf     bsmtexposure  housestyle  bsmtunfsf   lotarea
##      0.31654464  0.27827562  0.26080649  0.23979075  0.20994846
## condition1     bedroomabvgr  salecondition  kitchenabvgr  functional
##      0.18742710  0.16286668  0.15899900  0.14185984  0.12634291
## overallcond     bsmtfinsf2
##      0.11069778  0.02201662
```

Some variables have more impact than others but nevertheless they are statistically significant in our model so we keep them. Three of these variables are generated from other variables. We created `summertime` partly because of common sense and after plotting the distribution of houses being sold by month, we saw a peak in the summer months. This makes sense practically because people tend to have more time during the

summer and thus are more likely to buy a house. Secondly, we created `soldminusbuilt` because we felt that the difference between `yearsold` and `yearbuilt` is more useful together rather than separately. The third variable we created is a boolean for `saletype` to indicate a house that was “just constructed and sold”, which from a common sense perspective, can make the house go much higher. Many of the variables are condensed into smaller levels. Many levels have very few observations so we feel they are not significant enough to have their own level. This helps to prevent overfitting when predicting new values. We chose to not have too many variables in our model to also prevent overfitting. We confirmed the validity of our variables through LASSO regression. Lasso didn’t really eliminate any variables, which supports the statistical significance of our predictors.