dong_chris_housing

Chris Dong

September 20, 2017

Loading the data and any packages

```
options("max.print"=3)
suppressMessages(library(tidyverse))
suppressMessages(library(magrittr))
suppressMessages(library(leaps))
suppressMessages(library(VIM))
suppressMessages(library(car))
suppressMessages(library(Hmisc))
house <- read_csv("housing.txt", col_types = cols())
names(house) <- tolower(names(house))
```

Convert `mssubclass` to factor and check for `NA`s

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

```
##      poolqc miscfeature       alley
##        1453        1406        1369
##  [ reached getOption("max.print") -- omitted 78 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$bsmtfintype1[which(is.na(house$bsmtfintype1))] <- 0
house$bsmtfintype2[which(is.na(house$bsmtfintype2))] <- 0
house$masvnrarea <- as.numeric(house$masvnrarea)
house$masvnrarea[which(is.na(house$masvnrarea))] <- 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 `NA`s 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 %<>% 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.

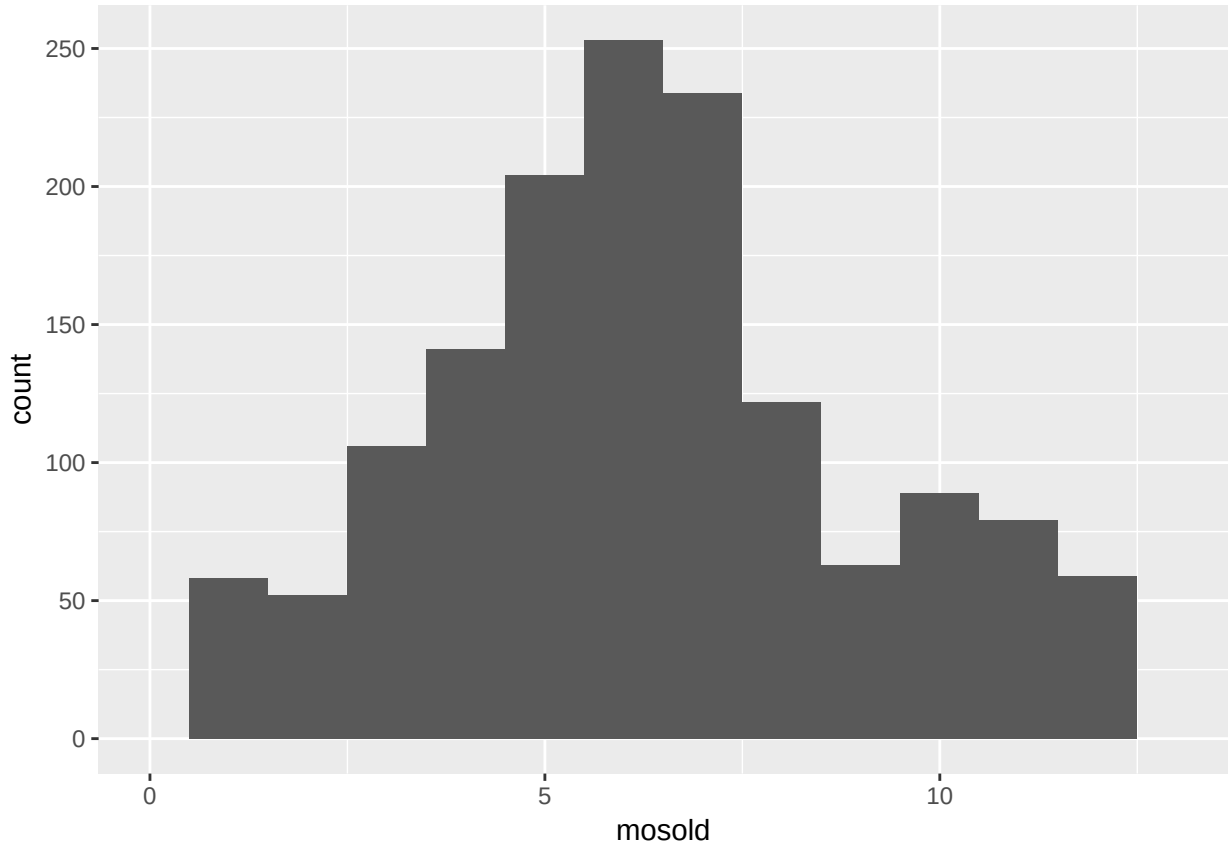
```
table(house$lotshape)
```

```
##
## 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)
```




```
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 signifiance for a particlar 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, I noticed `lotfrontage` became significant when I take the square root.

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

```
## [1] 0.9328122
```

```
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
##   1220     31     52
##  [ reached getOption("max.print") -- omitted 2 entries ]
```

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

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

```
## # A tibble: 6 x 2
##   salecondition avgprc
##           <chr>  <dbl>
## 1       Partial 244600
## 2        Normal 160000
## 3        Alloca 148145
## 4        Family 140500
## 5       Abnorml 130000
## 6       AdjLand 104000
```

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

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

```
## # A tibble: 9 x 2
##   saletype avgprc
##      <chr>  <dbl>
## 1      Con 269600
## 2      New 247453
## 3      CWD 188750
## 4       WD 158000
## 5    ConLw 144000
```

```
## 6    ConLD 140000
## 7      COD 139000
## 8    ConLI 125000
## 9      Oth 116050
```

```
house$newtype <- (house$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
##   1  49 428
##  [ reached getOption("max.print") -- omitted 2 entries ]
```

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

house %<>% select(-heating)

table(house$bsmtfintype1)
```

```
##
##   0 ALQ BLQ
##  37 220 148
##  [ reached getOption("max.print") -- omitted 4 entries ]
```

```
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
##   38  221  134
##  [ reached getOption("max.print") -- omitted 2 entries ]
```

```
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
##   45   65   37
##  [ reached getOption("max.print") -- omitted 2 entries ]
```

```
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))
```

```r
cor(house$bsmtcond,house$bsmtqual)
```

```
## [1] 0.6337134
```

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

```
## [1] 0.2126072
```

```
## [1] 0.5852072
```

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

table(house$foundation)
```

```
##
## BrkTil CBlock  PConc
##    146    634    647
##  [ reached getOption("max.print") -- omitted 3 entries ]
```

```r
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
```

```r
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
```

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

table(house$masvnrtype)
```

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

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

house_by_exterior <- house %>%  group_by(exterior1st) %>% summarise(avgprc = median(saleprice)) %>% arra

house_by_exterior$exteriorcategory <- as.numeric(factor(cut2(house_by_exterior$avgprc, quantile(house_by
       labels = 1:4))
```

```
house_by_exterior <- house_by_exterior[,-2]

house %<>% left_join(house_by_exterior, by = "exterior1st") %>% select(-exterior1st)

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

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

```
table(house$housestyle)
```

```
##
## 1.5Fin 1.5Unf 1Story
##    154     14    726
##  [ reached getOption("max.print") -- omitted 5 entries ]
```

```
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
##   1220     31     52
##  [ reached getOption("max.print") -- omitted 2 entries ]
```

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

```
## # A tibble: 5 x 2
##   bldgtype avgprc
##      <chr>  <dbl>
## 1   TwnhsE 172200
## 2     1Fam 167900
## 3    Twnhs 137500
## 4   Duplex 135980
## 5   2fmCon 127500
```

```
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
##      10      65      16
##  [ reached getOption("max.print") -- omitted 2 entries ]
```

```
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)


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$garagequal, house$garagecars)
```

```
## [1] 0.5766224
```

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

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

```
## [1] 0.8980772
```

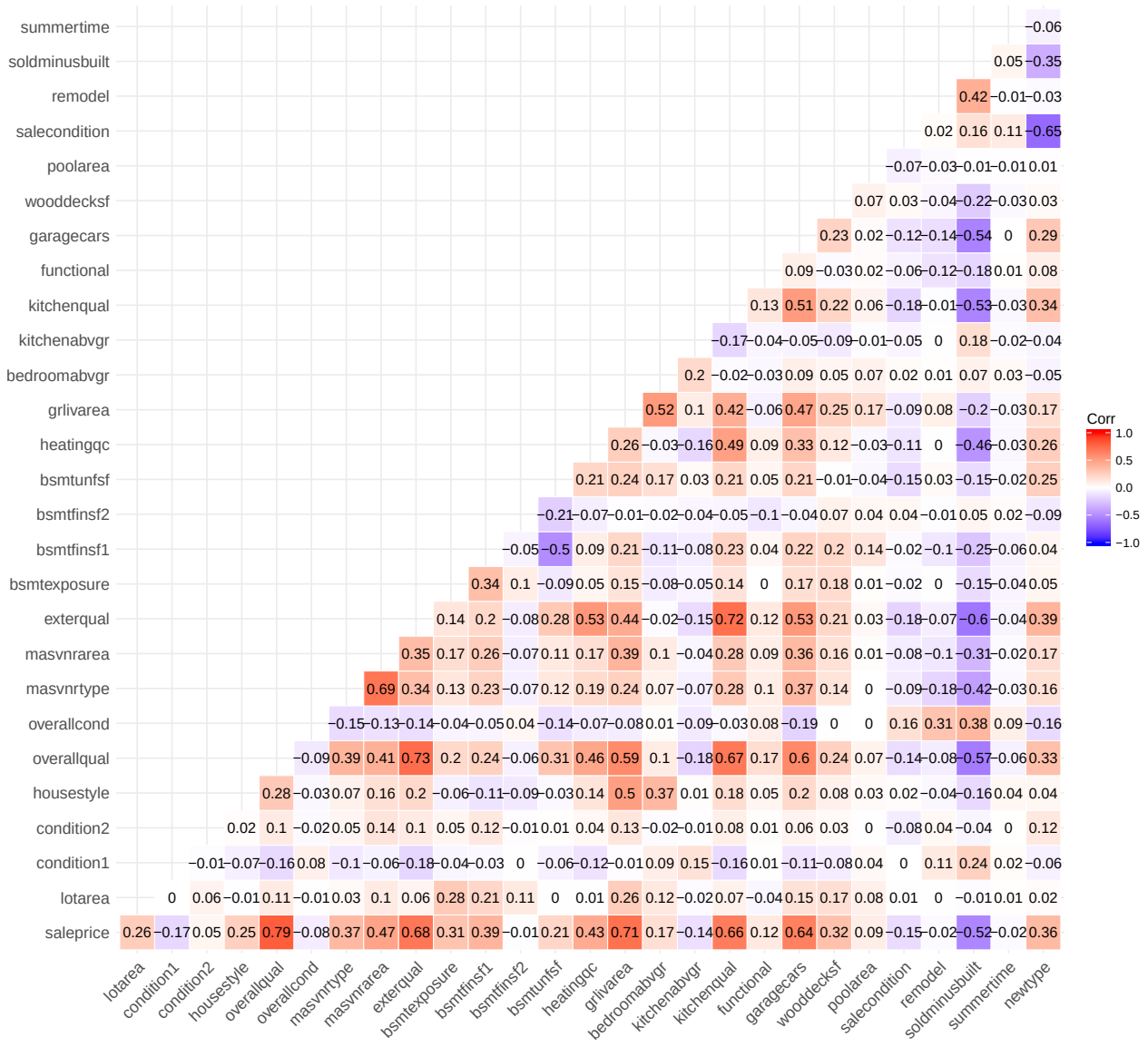
Checking multicollinearity. Looks good. For the generalized variance inflation factor (normalized by the degree of freedom), everything except one is less than 2.

```
vif(fullmodel)
```

```
##                     GVIF Df GVIF^(1/(2*Df))
## lotarea         1.433292  1        1.197202
##  [ reached getOption("max.print") -- omitted 29 rows ]
```

Getting all of the numeric variables.

```
house$remodel <- as.numeric(house$remodel)

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

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

library(ggcorrplot)

cor_matrix <- cor(house_numeric)

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



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

```
bestpredictors <- names(house_numeric)[sapply(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.

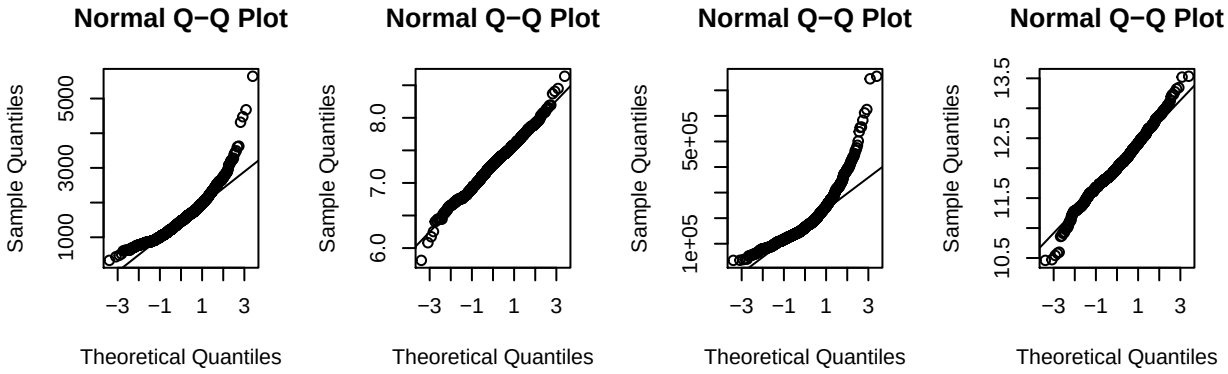
```
cor(housesubset)
```

```
##             overallqual exterqual grlivarea kitchenqual garagecars
##  [ reached getOption("max.print") -- omitted 5 rows ]
```

```
vif(bestmodel)
```

```
##                  GVIF Df GVIF^(1/(2*Df))
## overallqual  3.464742  1        1.861382
##  [ reached getOption("max.print") -- omitted 5 rows ]
```

```
par(mfrow=c(2,4))
qqnorm(housesubset$grlivarea); qqline(housesubset$grlivarea)
qqnorm(log(housesubset$grlivarea)); qqline(log(housesubset$grlivarea))

qqnorm(house$saleprice); qqline(house$saleprice)
qqnorm(log(house$saleprice)); qqline(log(house$saleprice))
```




```
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
## -0.97098 -0.07887  0.01184
##  [ reached getOption("max.print") -- omitted 2 entries ]
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
##  [ reached getOption("max.print") -- omitted 30 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 influence points

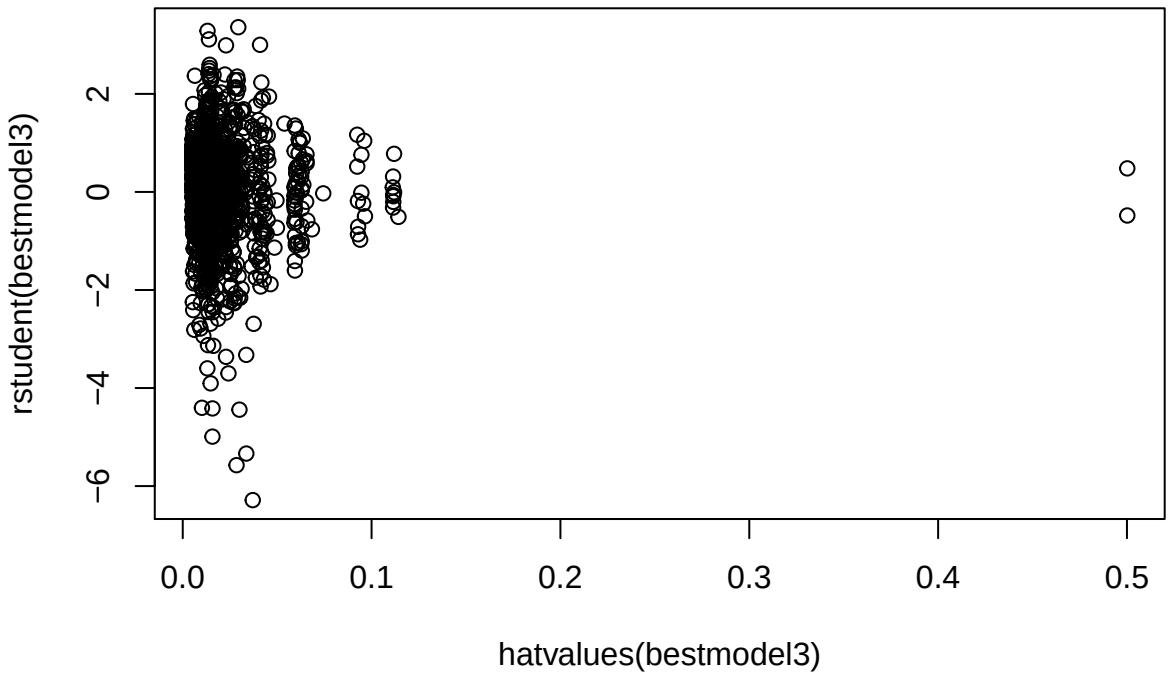
```
infm <- influence.measures(bestmodel3)
which(apply(infm$is.inf,1,any)) #influential observations
```

```
##  2  4 18
##  2  4 18
##  [ reached getOption("max.print") -- omitted 134 entries ]
```

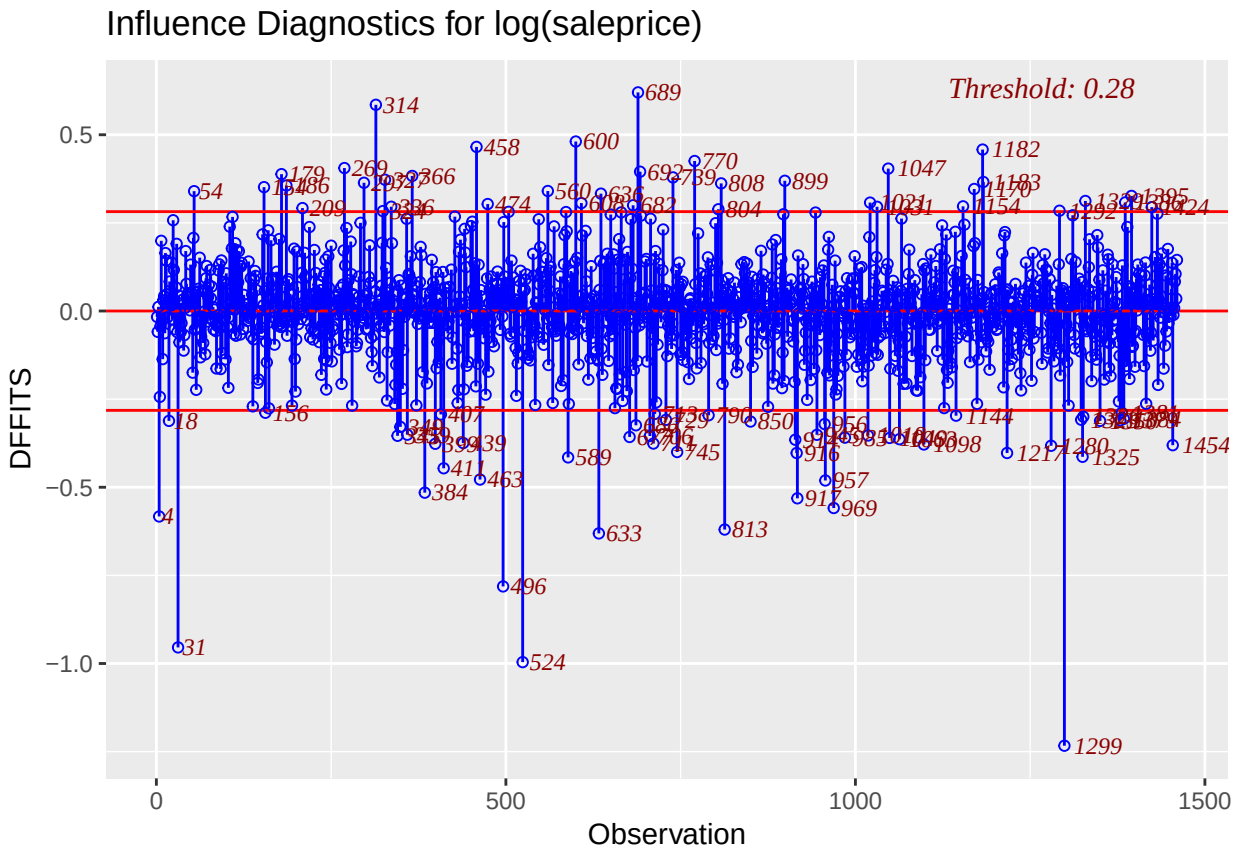
```
summary(infm)
```

```
## Potentially influential observations of
##   lm(formula = log(saleprice) ~ overallqual + log(grlivarea) +      kitchenqual + garagecars + neighl
##
##      dfb.1_ dfb.ovrl dfb.lg() dfb.ktch dfb.grgc dfb.nghB dfb.ngBD dfb.ngBS
##      dfb.nghbrhdClrC dfb.nghbrhdCllC dfb.nghC dfb.nghE dfb.nghG dfb.nIDO
##      dfb.ngMV dfb.nghM dfb.ngNA dfb.ngNR dfb.nNPV dfb.ngNH dfb.nNWA
##      dfb.ngOT dfb.nghbrhdSw dfb.ngSW dfb.nghbrhdSm dfb.ngSB dfb.nSWI
##      dfb.nghT dfb.nghV dffit   cov.r   cook.d hat
##  [ reached getOption("max.print") -- omitted 137 rows ]
```

```
plot(rstudent(bestmodel3) ~ hatvalues(bestmodel3))
```




```
#install.packages("olsrr")
suppressMessages(library(olsrr))
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

Let's see what happens if we simply remove the observations.

```
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.

```
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] 0.866407
```

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.

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

```
## [1] 0.9251491
```

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] 0.923607
```

We can try removing the outliers, which improved the R squared by a lot. Now, we can test some interaction terms.

```
model31varremove <- lm(log(saleprice) ~ log(lotarea) +
              sqrt(bsmtfinsf1)+log(grlivarea)+., data = house2[-influenceindex,])
summary(model31varremove)$r.squared
```

```
## [1] 0.9469088
```

I remove some variables found to be insignificant.

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

I look back at the correlation plot generated earlier and tested random interaction terms. I found the interaction of `overallqual` and `grlivarea` to be significant.

```
modelinteraction <- lm(log(saleprice) ~ log(lotarea) +
              sqrt(bsmtfinsf1)+log(grlivarea) +. -
                lotarea - bsmtfinsf1 - grlivarea,
              data = house3[-influenceindex,])
summary(modelinteraction)$r.squared
```

```
## [1] 0.9439097
```

```
vif(modelinteraction)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## log(lotarea)      2.455356  1        1.566958
##  [ reached getOption("max.print") -- omitted 24 rows ]
```

Reduce the multicollinearity due to interaction terms by standardizing the variables.

Remove `exterqual`

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

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,])
vif(endmodel)
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## log(lotarea)      2.454861  1        1.566800
##  [ reached getOption("max.print") -- omitted 23 rows ]
```

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

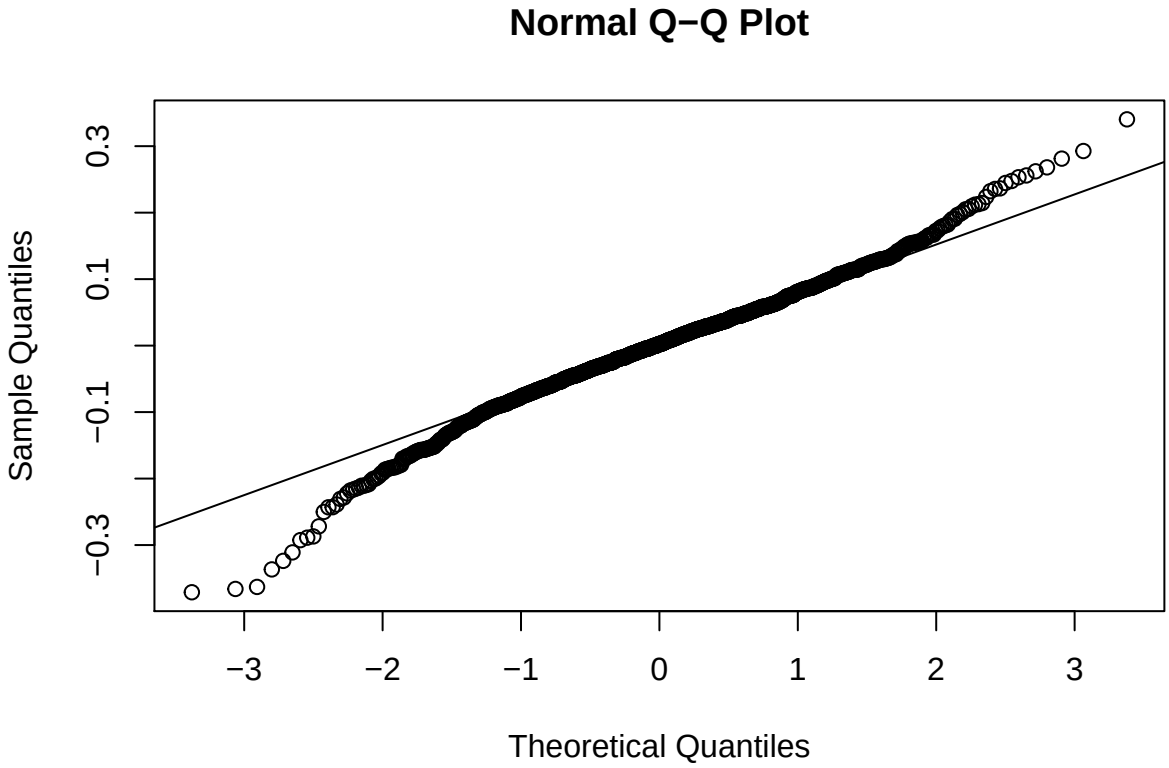
```
##
## Call:
## lm(formula = log(saleprice) ~ log(lotarea) + sqrt(bsmtfinsf1) +
##     log(grlivarea) + . - lotarea - bsmtfinsf1 - grlivarea, data = house4[-influenceindex,
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.37097 -0.04956  0.00242  0.05213  0.34034
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          6.975e+00  1.090e-01  63.999  < 2e-16 ***
## log(lotarea)         9.534e-02  7.768e-03  12.275  < 2e-16 ***
## sqrt(bsmtfinsf1)     4.981e-03  3.238e-04  15.383  < 2e-16 ***
## log(grlivarea)       4.584e-01  1.581e-02  28.996  < 2e-16 ***
## neighborhoodBrDale  -6.989e-02  3.586e-02  -1.949 0.051482 .
## neighborhoodBrkSide -8.722e-04  2.997e-02  -0.029 0.976786
## neighborhoodClearCr  3.361e-02  3.367e-02   0.998 0.318384
## neighborhoodCollgCr -5.904e-03  2.625e-02  -0.225 0.822088
## neighborhoodCrawfor  1.202e-01  3.051e-02   3.939 8.62e-05 ***
## neighborhoodEdwards -7.077e-02  2.834e-02  -2.497 0.012656 *
## neighborhoodGilbert -9.594e-03  2.788e-02  -0.344 0.730840
## neighborhoodIDOTRR  -8.905e-02  3.337e-02  -2.669 0.007704 **
## neighborhoodMeadowV -7.746e-02  3.486e-02  -2.222 0.026442 *
## neighborhoodMitchel -3.302e-02  2.929e-02  -1.127 0.259840
```

```
## neighborhoodNAmes   -2.774e-02  2.737e-02  -1.014 0.310966
## neighborhoodNoRidge  6.854e-02  3.052e-02   2.246 0.024890 *
## neighborhoodNPkVill  3.474e-03  3.923e-02   0.089 0.929444
## neighborhoodNridgHt  8.628e-02  2.731e-02   3.159 0.001619 **
## neighborhoodNWAmes  -3.035e-02  2.847e-02  -1.066 0.286500
## neighborhoodOldTown -7.253e-02  2.933e-02  -2.472 0.013543 *
## neighborhoodSawyer   7.942e-04  2.896e-02   0.027 0.978128
## neighborhoodSawyerW -2.200e-02  2.826e-02  -0.778 0.436435
## neighborhoodSomerst  6.144e-02  2.649e-02   2.319 0.020543 *
## neighborhoodStoneBr  1.185e-01  3.336e-02   3.553 0.000394 ***
## neighborhoodSWISU   -4.328e-02  3.429e-02  -1.262 0.207104
## neighborhoodTimber   2.582e-03  2.973e-02   0.087 0.930826
## neighborhoodVeenker  1.816e-02  4.149e-02   0.438 0.661642
## condition1          -6.237e-02  8.700e-03  -7.169 1.25e-12 ***
## housestyle          -2.119e-02  8.011e-03  -2.646 0.008249 **
## overallqual          5.538e-02  3.509e-03  15.781  < 2e-16 ***
## overallcond          3.705e-02  2.729e-03  13.576  < 2e-16 ***
## masvnrtype          -1.518e-02  7.595e-03  -1.998 0.045887 *
## masvnrarea           5.105e-05  2.120e-05   2.408 0.016185 *
## bsmtexposure         5.005e-02  9.667e-03   5.178 2.59e-07 ***
## bsmtfinsf2           7.966e-05  1.662e-05   4.793 1.83e-06 ***
## bsmtunfsf            6.728e-05  9.757e-06   6.896 8.29e-12 ***
## heatingqc            2.437e-02  6.208e-03   3.926 9.09e-05 ***
## bedroomabvgr        -1.295e-02  4.229e-03  -3.062 0.002246 **
## kitchenabvgr        -5.529e-02  1.282e-02  -4.312 1.74e-05 ***
## kitchenqual          4.114e-02  5.649e-03   7.283 5.58e-13 ***
## functional           7.798e-02  1.082e-02   7.208 9.50e-13 ***
## garagecars           4.742e-02  4.780e-03   9.921  < 2e-16 ***
## wooddecksf           7.994e-05  2.142e-05   3.731 0.000199 ***
## salecondition        4.426e-02  8.798e-03   5.030 5.57e-07 ***
## soldminusbuilt      -2.112e-03  2.101e-04 -10.050  < 2e-16 ***
## summertime           1.784e-02  4.905e-03   3.638 0.000285 ***
## newtype              1.042e-01  1.323e-02   7.881 6.74e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08875 on 1324 degrees of freedom
## Multiple R-squared:  0.9439, Adjusted R-squared:  0.9419
## F-statistic:   484 on 46 and 1324 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 = 0.036643, p-value = 0.05036
## alternative hypothesis: two-sided
```

```
qqnorm(endmodel$residuals); qqline(endmodel$residuals)
```



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" "summertime"
## [25] "newtype"
```

```
signif_var <- house4 %>% select(-neighborhood) %>%
  sapply(function(x) abs(cor(house4$saleprice, x)))
signif_var[signif_var >= 0.5]
```

```
##    overallqual      grlivarea    kitchenqual     garagecars      saleprice
##      0.8131930      0.7019635      0.6832550      0.6635628      1.0000000
## soldminusbuilt
##      0.5646160
```

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()

```
transform <- function(df){
  names(morty) <- tolower(names(morty))
  morty$soldminusbuilt <- (morty$yrsold - morty$yearbuilt)
  morty$summertime <- (morty$mosold %in% 5:7) * 1
  morty$newtype <- (morty$saletype == 'New') * 1

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

  morty$condition1 <- (morty$condition1 == "Artery" |
      morty$condition1 =="Feedr"| morty$condition1 == "RRAe")*1


  return(morty)
}
transform(morty)
```

```
## # A tibble: 1 x 25
##   lotarea neighborhood condition1 housestyle overallqual overallcond
##     <int>        <chr>      <dbl>      <chr>       <int>       <int>
## 1   14115      Mitchel          0     1.5Fin           5           5
## # ... with 19 more variables: masvnrtype <chr>, masvnrarea <int>,
## #   bsmtexposure <chr>, bsmtfinsf1 <int>, bsmtfinsf2 <int>,
## #   bsmtunfsf <int>, heatingqc <chr>, grlivarea <int>, bedroomabvgr <int>,
## #   kitchenabvgr <int>, kitchenqual <chr>, functional <chr>,
## #   garagecars <int>, wooddecksf <int>, salecondition <chr>,
## #   saleprice <int>, soldminusbuilt <int>, summertime <dbl>, newtype <dbl>
```