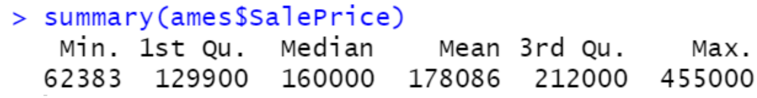
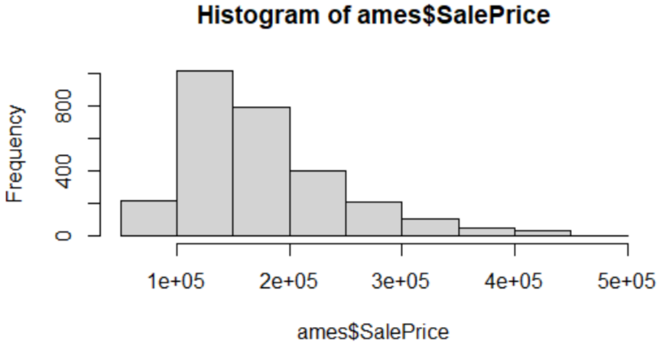
**Tim Miller**
**15.071 - Homework #1**

## Problem 1a



```
> summary(ames$SalePrice)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  62383  129900  160000  178086  212000  455000 
```

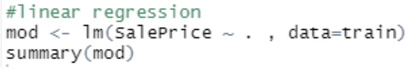
```
> hist(ames$SalePrice)
```



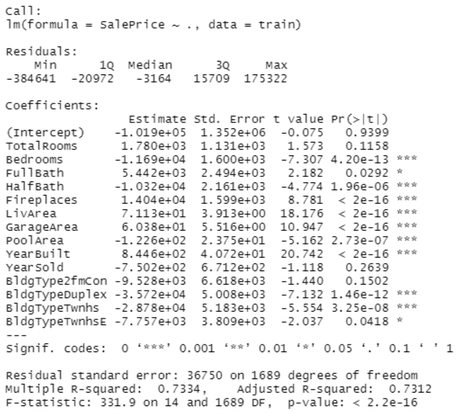
The distribution is right-skewed. This aligns with my intuition. If I think about my own town – most houses are the same size and price. There are also a scattering of larger houses and one or two mansions.

Initially I was surprised with a max price of $455K. But thinking more – this makes sense. Ames real estate is not expensive, given the location. Plus, large houses ~$1M are not likely sold that often.

## Problem 1b



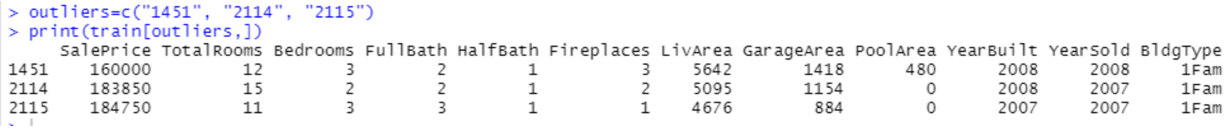
```
#linear regression
mod <- lm(SalePrice ~ . , data=train)
summary(mod)
```



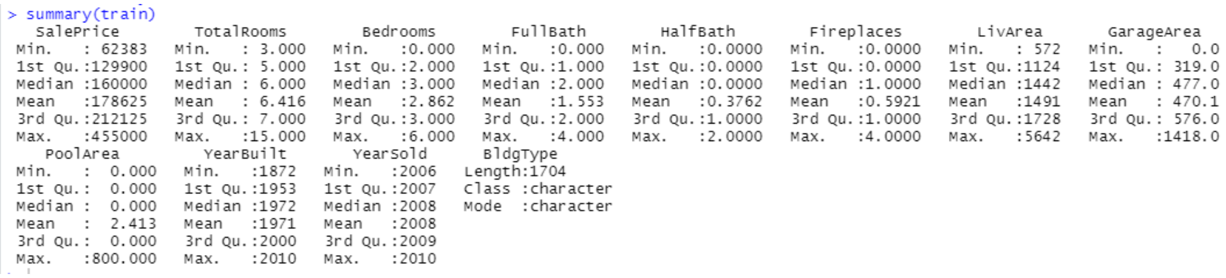
```
Call:
lm(formula = SalePrice ~ ., data = train)

Residuals:
    Min      1Q  Median      3Q     Max 
-384641  -20972   -3164   15709  175322 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -1.019e+05  1.352e+06  -0.075   0.9399    
TotalRooms      1.780e+03  1.131e+03   1.573   0.1158    
Bedrooms       -1.169e+04  1.600e+03  -7.307 4.20e-13 ***
FullBath        5.442e+03  2.494e+03   2.182   0.0292 *  
HalfBath       -1.032e+04  2.161e+03  -4.774 1.96e-06 ***
Fireplaces      1.404e+04  1.599e+03   8.781  < 2e-16 ***
LivArea         7.113e+01  3.913e+00  18.176  < 2e-16 ***
GarageArea      6.038e+01  5.516e+00  10.947  < 2e-16 ***
PoolArea       -1.226e+02  2.375e+01  -5.162 2.73e-07 ***
YearBuilt       8.446e+02  4.072e+01  20.742  < 2e-16 ***
YearSold       -7.502e+02  6.712e+02  -1.118   0.2639    
BldgType2fmCon -9.528e+03  6.618e+03  -1.440   0.1502    
BldgTypeDuplex -3.572e+04  5.008e+03  -7.132 1.46e-12 ***
BldgTypeTwnhs  -2.878e+04  5.183e+03  -5.554 3.25e-08 ***
BldgTypeTwnhsE -7.757e+03  3.809e+03  -2.037   0.0418 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 36750 on 1689 degrees of freedom
Multiple R-squared:  0.7334,	Adjusted R-squared:  0.7312 
F-statistic: 331.9 on 14 and 1689 DF,  p-value: < 2.2e-16
```

$R^2$ of this model is 0.7334.

## Problem 1c



```
> outliers=c("1451", "2114", "2115")
> print(train[outliers,])
     SalePrice TotalRooms Bedrooms FullBath HalfBath Fireplaces LivArea GarageArea PoolArea YearBuilt YearSold BldgType
1451    160000         12        3        2        1          3    5642       1418      480      2008     2008     1Fam
2114    183850         15        2        2        1          2    5095       1154        0      2008     2007     1Fam
2115    184750         11        3        3        1          1    4676        884        0      2007     2007     1Fam
```



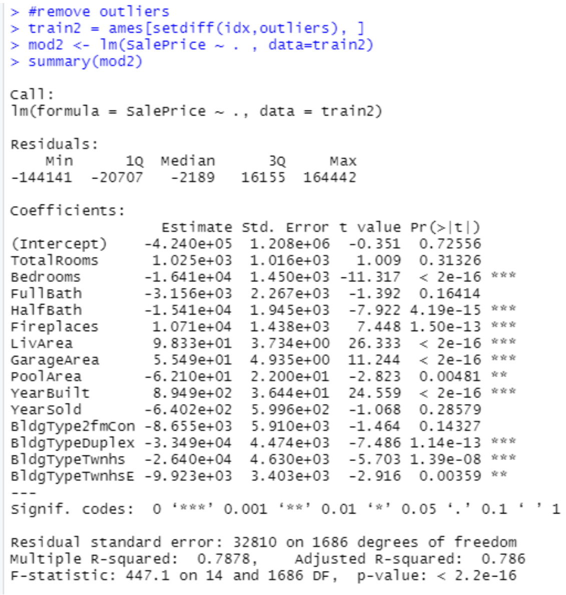
```
> summary(train)
   SalePrice        TotalRooms        Bedrooms        FullBath        HalfBath        Fireplaces        LivArea        GarageArea    
 Min.   : 62383   Min.   : 3.000   Min.   :0.000   Min.   :0.000   Min.   :0.0000   Min.   :0.0000   Min.   : 572   Min.   :   0.0  
 1st Qu.:129900   1st Qu.: 5.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:1124   1st Qu.: 319.0  
 Median :160000   Median : 6.000   Median :3.000   Median :2.000   Median :0.0000   Median :1.0000   Median :1442   Median : 477.0  
 Mean   :178625   Mean   : 6.416   Mean   :2.862   Mean   :1.553   Mean   :0.3762   Mean   :0.5921   Mean   :1491   Mean   : 470.1  
 3rd Qu.:212125   3rd Qu.: 7.000   3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1728   3rd Qu.: 576.0  
 Max.   :455000   Max.   :15.000   Max.   :6.000   Max.   :4.000   Max.   :2.0000   Max.   :4.0000   Max.   :5642   Max.   :1418.0  
    PoolArea          YearBuilt       YearSold      BldgType        
 Min.   :  0.000   Min.   :1872   Min.   :2006   Length:1704       
 1st Qu.:  0.000   1st Qu.:1953   1st Qu.:2007   Class :character  
 Median :  0.000   Median :1972   Median :2008   Mode  :character  
 Mean   :  2.413   Mean   :1971   Mean   :2008                     
 3rd Qu.:  0.000   3rd Qu.:2000   3rd Qu.:2009                     
 Max.   :800.000   Max.   :2010   Max.   :2010                     
```

For the outliers, TotalRooms, LivArea, and GarageArea columns are different than the rest of the observed values:

- TotalRooms: the three outliers have a minimum ~2x the number of mean rooms
- LivArea: the three outliers have a minimum ~3x living area
- GarageArea: the three outliers have a minimum ~2x garage area

In a normal, year the outlier values are realistic. But we have to consider YearBuilt column. We see that the three of these outliers were built at the height of the financial crisis. Likely, the developer built these large houses assuming they could ride the house price bull market. When housing prices collapsed, the owner / builder had no choice but to sell at a discounted price.

## Problem 1d



```
> #remove outliers
> train2 = ames[setdiff(idx,outliers), ]
> mod2 <- lm(SalePrice ~ . , data=train2)
> summary(mod2)

Call:
lm(formula = SalePrice ~ ., data = train2)

Residuals:
    Min      1Q  Median      3Q     Max 
-144141  -20707   -2189   16155  164442 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -4.240e+05  1.208e+06  -0.351  0.72556    
TotalRooms      1.025e+03  1.016e+03   1.009  0.31326    
Bedrooms       -1.641e+04  1.450e+03 -11.317  < 2e-16 ***
FullBath       -3.156e+03  2.267e+03  -1.392  0.16414    
HalfBath       -1.541e+04  1.945e+03  -7.922 4.19e-15 ***
Fireplaces      1.071e+04  1.438e+03   7.448 1.50e-13 ***
LivArea         9.833e+01  3.734e+00  26.333  < 2e-16 ***
GarageArea      5.549e+01  4.935e+00  11.244  < 2e-16 ***
PoolArea       -6.210e+01  2.200e+01  -2.823  0.00481 ** 
YearBuilt       8.949e+02  3.644e+01  24.559  < 2e-16 ***
YearSold       -6.402e+02  5.996e+02  -1.068  0.28579    
BldgType2fmCon -8.655e+03  5.910e+03  -1.464  0.14327    
BldgTypeDuplex -3.349e+04  4.474e+03  -7.486 1.14e-13 ***
BldgTypeTwnhs  -2.640e+04  4.630e+03  -5.703 1.39e-08 ***
BldgTypeTwnhsE -9.923e+03  3.403e+03  -2.916  0.00359 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 32810 on 1686 degrees of freedom
Multiple R-squared:  0.7878,	Adjusted R-squared:  0.786 
F-statistic: 447.1 on 14 and 1686 DF,  p-value: < 2.2e-16
```

After removing the outliers, we find the $R^2$ increase from 0.7334 to 0.7878. This signals a better fit for the model.

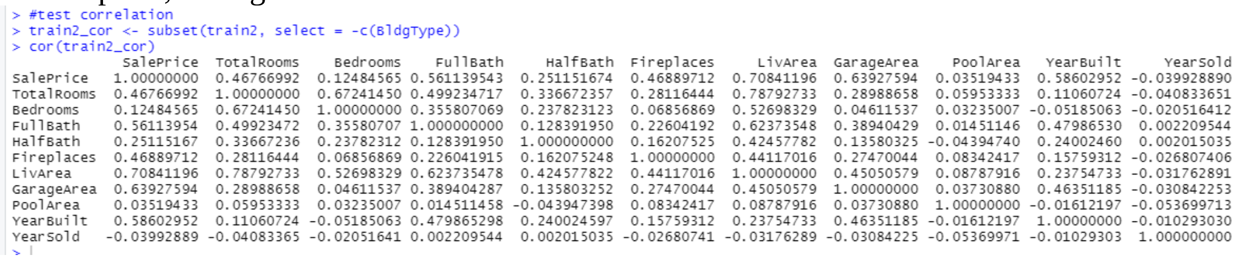
## Problem 1e

This is not a robust model - we included all variables. Some unexpected results are not surprising.

That said, I want to investigate negative coefficients for Bedrooms, FullBath, HalfBath, and PoolArea. My intuition tells me that home prices increases with more bedrooms, bathrooms, and a pool.

Furthermore, I would expect statistical significance for TotalRooms and FullBath variables. For the former, this is a proxy for house size: larger houses generally are priced higher. But there could be some correlation with LivArea causing issues.

To that point, testing variable correlation:



```
> #test correlation
> train2_cor <- subset(train2, select = -c(BldgType))
> cor(train2_cor)
             SalePrice  TotalRooms    Bedrooms    FullBath     HalfBath  Fireplaces     LivArea  GarageArea    PoolArea   YearBuilt     YearSold
SalePrice   1.00000000  0.46766992  0.12484565 0.561139543  0.251151674  0.46889712  0.70841196  0.63927594  0.03519433  0.58602952 -0.039928890
TotalRooms  0.46766992  1.00000000  0.67241450 0.499234717  0.336672357  0.28116444  0.78792733  0.28988658  0.05953333  0.11060724 -0.040833651
Bedrooms    0.12484565  0.67241450  1.00000000 0.355807069  0.237823123  0.06856869  0.52698329  0.04611537  0.03235007 -0.05185063 -0.020516412
FullBath    0.56113954  0.49923472  0.35580707 1.000000000  0.128391950  0.22604192  0.62373548  0.38940429  0.01451146  0.47986530  0.002209544
HalfBath    0.25115167  0.33667236  0.23782312 0.128391950  1.000000000  0.16207525  0.42457782  0.13580325 -0.04394740  0.24002460  0.002015035
Fireplaces  0.46889712  0.28116444  0.06856869 0.226041915  0.162075248  1.00000000  0.44117016  0.27470044  0.08342417  0.15759312 -0.026807406
LivArea     0.70841196  0.78792733  0.52698329 0.623735478  0.424577822  0.44117016  1.00000000  0.45050579  0.08787916  0.23754733 -0.031762891
GarageArea  0.63927594  0.28988658  0.04611537 0.389404287  0.135803252  0.27470044  0.45050579  1.00000000  0.03730880  0.46351185 -0.030842253
PoolArea    0.03519433  0.05953333  0.03235007 0.014511458 -0.043947398  0.08342417  0.08787916  0.03730880  1.00000000 -0.01612197 -0.053699713
YearBuilt   0.58602952  0.11060724 -0.05185063 0.479865298  0.240024597  0.15759312  0.23754733  0.46351185 -0.01612197  1.00000000 -0.010293030
YearSold   -0.03992889 -0.04083365 -0.02051641 0.002209544  0.002015035 -0.02680741 -0.03176289 -0.03084225 -0.05369971 -0.01029303  1.000000000
> 
```

We relatively significant correlation between:

- TotalRooms <> Bedrooms (0.67)
- TotalRooms <> LivArea (0.78)
- FullBath <> LivArea (0.62)

The correlation of these variables is likely driving the unexpected coefficient signs. As a next step we would likely drop TotalRooms from the model since it has such high correlation with LivArea (plus LivArea is more highly correlated with SalePrice).

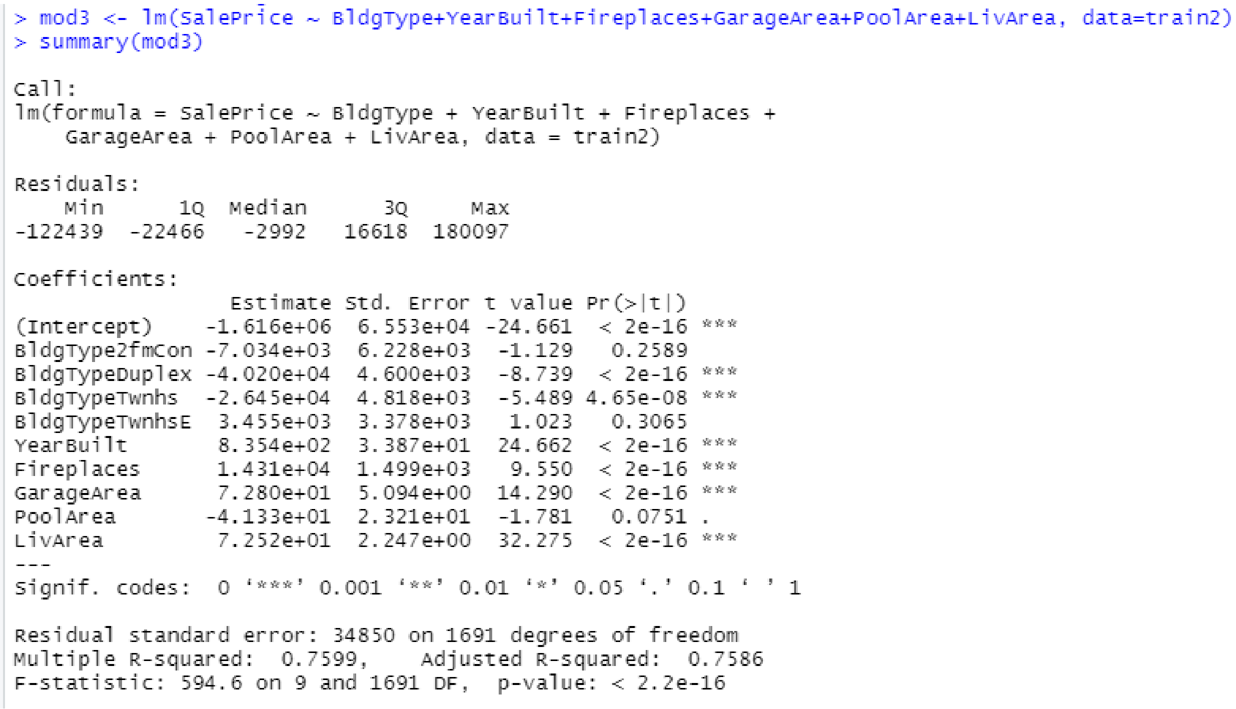
## Problem 1f

Dependent variables values in regressions models are based on the interaction of numerous independent variables. This model cannot support the fireplace claim because simply adding a fireplace does not consider the interaction with all the other variables. Intuitively, if you took away some livable area to install the fireplace, this could actually reduce the price, since livable area is also an important input into the sale price.

We can rephrase to:

- Considering the interaction of fireplaces with all other important inputs into house prices, the number of fireplaces is positively correlated with higher house prices.

# Problem 1g



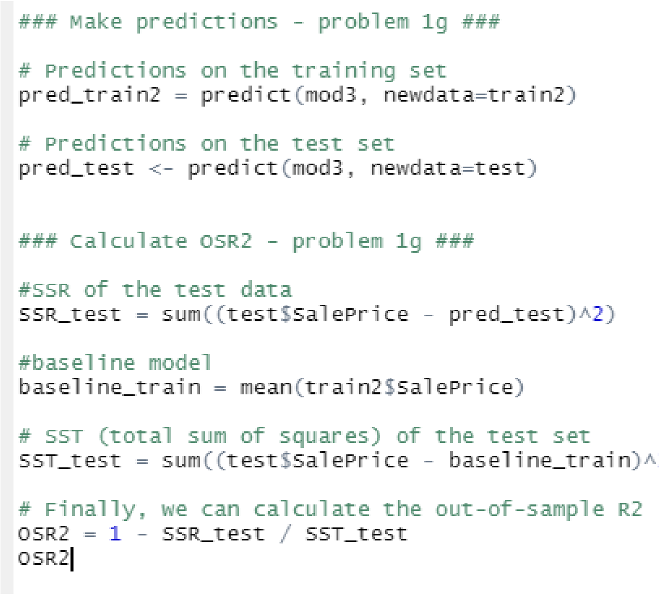
```
> mod3 <- lm(SalePrice ~ BldgType+YearBuilt+Fireplaces+GarageArea+PoolArea+LivArea, data=train2)
> summary(mod3)

Call:
lm(formula = SalePrice ~ BldgType + YearBuilt + Fireplaces + 
    GarageArea + PoolArea + LivArea, data = train2)

Residuals:
    Min      1Q  Median      3Q     Max 
-122439  -22466   -2992   16618  180097 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -1.616e+06  6.553e+04 -24.661  < 2e-16 ***
BldgType2fmCon -7.034e+03  6.228e+03  -1.129   0.2589    
BldgTypeDuplex -4.020e+04  4.600e+03  -8.739  < 2e-16 ***
BldgTypeTwnhs  -2.645e+04  4.818e+03  -5.489 4.65e-08 ***
BldgTypeTwnhsE  3.455e+03  3.378e+03   1.023   0.3065    
YearBuilt       8.354e+02  3.387e+01  24.662  < 2e-16 ***
Fireplaces      1.431e+04  1.499e+03   9.550  < 2e-16 ***
GarageArea      7.280e+01  5.094e+00  14.290  < 2e-16 ***
PoolArea       -4.133e+01  2.321e+01  -1.781   0.0751 .  
LivArea         7.252e+01  2.247e+00  32.275  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 34850 on 1691 degrees of freedom
Multiple R-squared:  0.7599,	Adjusted R-squared:  0.7586 
F-statistic: 594.6 on 9 and 1691 DF,  p-value: < 2.2e-16
```

Compared to the model in Problem 1d, we find the $R^2$ decrease from 0.7878 to 0.7599.

*$OSR^2$*

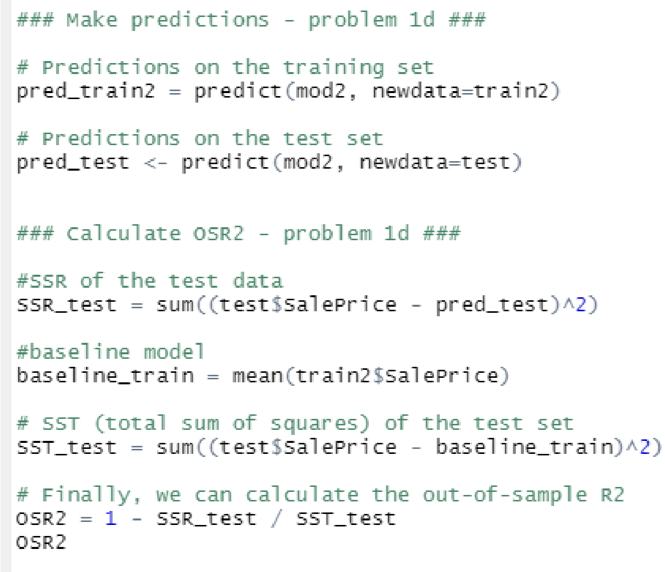
For the model in Problem 1g, $OSR^2$ = 0.7449623. This is less than the $OSR^2$ of 0.7715383 for the model in Problem 1d. Code below.

- $OSR^2$ for the model in Problem 1g



```
### Make predictions - problem 1g ###

# Predictions on the training set
pred_train2 = predict(mod3, newdata=train2)

# Predictions on the test set
pred_test <- predict(mod3, newdata=test)


### Calculate OSR2 - problem 1g ###

#SSR of the test data
SSR_test = sum((test$SalePrice - pred_test)^2)

#baseline model
baseline_train = mean(train2$SalePrice)

# SST (total sum of squares) of the test set
SST_test = sum((test$SalePrice - baseline_train)^2)

# Finally, we can calculate the out-of-sample R2
OSR2 = 1 - SSR_test / SST_test
OSR2
```

- OSR$^2$ for the model in Problem 1d

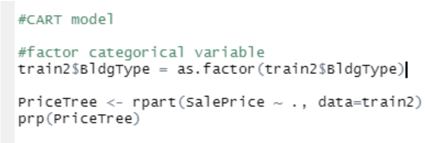


```
### Make predictions - problem 1d ###

# Predictions on the training set
pred_train2 = predict(mod2, newdata=train2)

# Predictions on the test set
pred_test <- predict(mod2, newdata=test)


### Calculate OSR2 - problem 1d ###

#SSR of the test data
SSR_test = sum((test$SalePrice - pred_test)^2)

#baseline model
baseline_train = mean(train2$SalePrice)

# SST (total sum of squares) of the test set
SST_test = sum((test$SalePrice - baseline_train)^2)

# Finally, we can calculate the out-of-sample R2
OSR2 = 1 - SSR_test / SST_test
OSR2
```

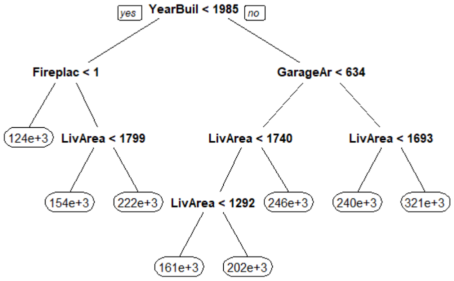
I would use the model from Problem 1g to both predict price house prices and analyze the relationship between different features and house price. The reason: it is a simpler model that removes variables that are correlated. R$^2$ for the Problem 1d is not significantly larger. Therefore, I am fine sacrificing some R$^2$ for a model that does not have significant correlation between variables.

## Problem 2a

Based on the model, the important variables are YearBuilt, Fireplace, LivArea, GarageArea. Somewhat surprised that number of bedrooms or bathrooms are in the tree. But my guess is that the model encompasses those variables in LivArea. Furthermore, would not have expected Fireplaces to be as significant of a variable – but that probably is an indicator for number of rooms and overall house size which should drive home price. But overall, the model is intuitive and sensible.

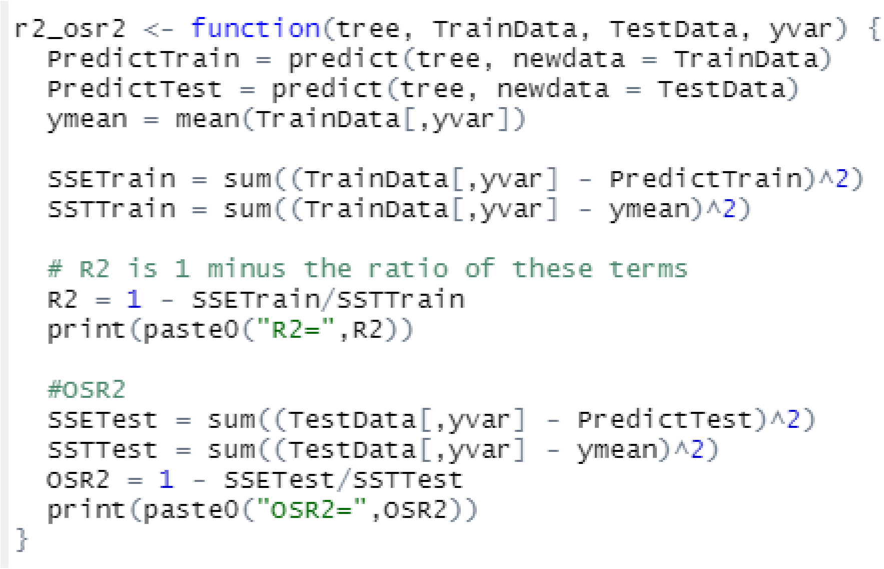


```
#CART model

#factor categorical variable
train2$BldgType = as.factor(train2$BldgType)

PriceTree <- rpart(SalePrice ~ ., data=train2)
prp(PriceTree)
```



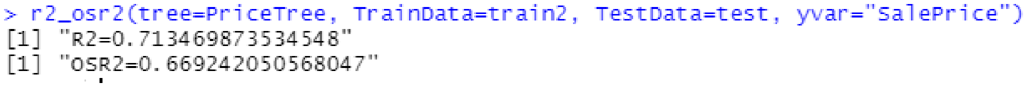
## Problem 2b

Setting up the function



```
r2_osr2 <- function(tree, TrainData, TestData, yvar) {
  PredictTrain = predict(tree, newdata = TrainData)
  PredictTest = predict(tree, newdata = TestData)
  ymean = mean(TrainData[,yvar])

  SSETrain = sum((TrainData[,yvar] - PredictTrain)^2)
  SSTTrain = sum((TrainData[,yvar] - ymean)^2)

  # R2 is 1 minus the ratio of these terms
  R2 = 1 - SSETrain/SSTTrain
  print(paste0("R2=",R2))

  #OSR2
  SSETest = sum((TestData[,yvar] - PredictTest)^2)
  SSTTest = sum((TestData[,yvar] - ymean)^2)
  OSR2 = 1 - SSETest/SSTTest
  print(paste0("OSR2=",OSR2))
}
```

Executing the function

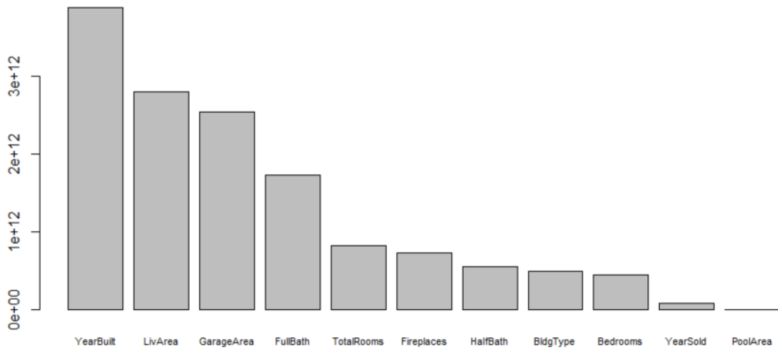


```
> r2_osr2(tree=PriceTree, TrainData=train2, TestData=test, yvar="SalePrice")
[1] "R2=0.713469873534548"
[1] "OSR2=0.669242050568047"
```

We find $R^2$ = 0.713469873534548. The $R^2$ for the model in 1g was 0.7599. So compared to that, the $R^2$ for the CART model is less. But not significantly so.

## Problem 2c

See separate file for print out of large tree.



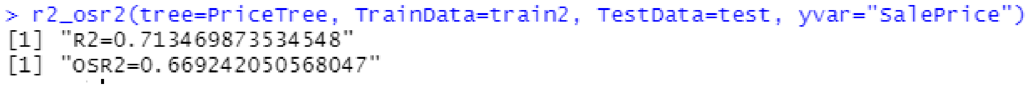
The five most important variables with the CART model are:

- YearBuilt
- LivArea
- GarageArea
- Fullbath
- TotalRooms

This aligns well with linear regression. Fireplaces is the one variable that is relatively highly correlated with SalePrice that does not appear in the top 5 of the CART model. Furthermore, BldgType does not contribute very much in the CART model, but we did see it was statistically significant in the model from problem 1g.

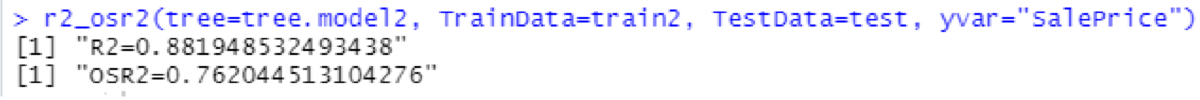
## Problem 2d

For the first model, executing the function we set up in Problem 2b:



```
> r2_osr2(tree=PriceTree, TrainData=train2, TestData=test, yvar="SalePrice")
[1] "R2=0.713469873534548"
[1] "OSR2=0.669242050568047"
```

For the second model, executing the function we set up in Problem 2b:



```
> r2_osr2(tree=tree.model2, TrainData=train2, TestData=test, yvar="SalePrice")
[1] "R2=0.881948532493438"
[1] "OSR2=0.762044513104276"
```

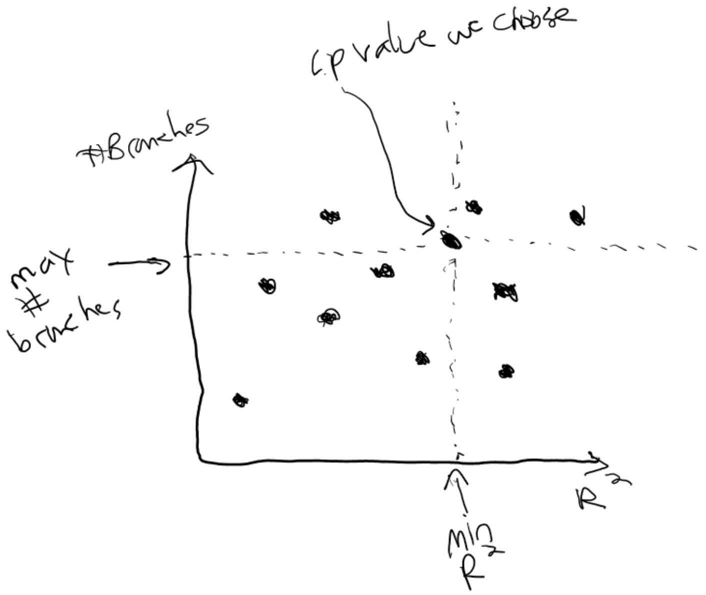
For the first model the OSR$^2$ is 0.669242050568047.  For the second model the OSR$^2$ is 0.762044513104276. Given the tree in the second model has more branches, it is no surprise that is has a higher OSR$^2$. But the model is likely overfitted and would be too complicated to communicate to any audience. So yes, it is more precise and might help with more accurate predictions, but the risk of overfitting is too high. If I were trying to make predictions, I would just use the first model.

## Problem 2e

The best CART model (model #2) has a higher R$^2$ and OSR$^2$ than both the linear regression models. But as I stated in Problem 2d, the best CART model has way too many branches. It risks overfitting and would be hard to communicate to any stakeholder. Plus the R$^2$ of the best CART model is not significantly better than the regression. Therefore, for sake of simplicity and practicality, I would use the regression model if I was trying to make predictions.

## Problem 2f

For a specific cp value I would plot number of branches vs. $R^2$ value of the CART model. Then I would have in mind a practical number of branches and a "good" enough $R^2$ value. I would pick the cp value that is at the intersection. The idea here is that we want to maximize $R^2$ but we do not want to have too many branches in the tree that it is hard to communicate to someone. Below is a sample of the type of chart I would create:



## Problem 2g

We need to split data into test and training sets so that we can confirm the applicability of the training model. Essentially we are saying that the test and the training data are random samples. So the model made using the training data should perform similarly well with the test data.

The issues that could arise are that the training and test data sets are not actually similar. So what happens is that the model on the trained data won't actually be that representative. The problem with selecting $R^2$ and comparing $OSR^2$ is that we are overfitting the data for this sample of the test and training data. And it might not actually be representative for the entire universe of potential outcomes.