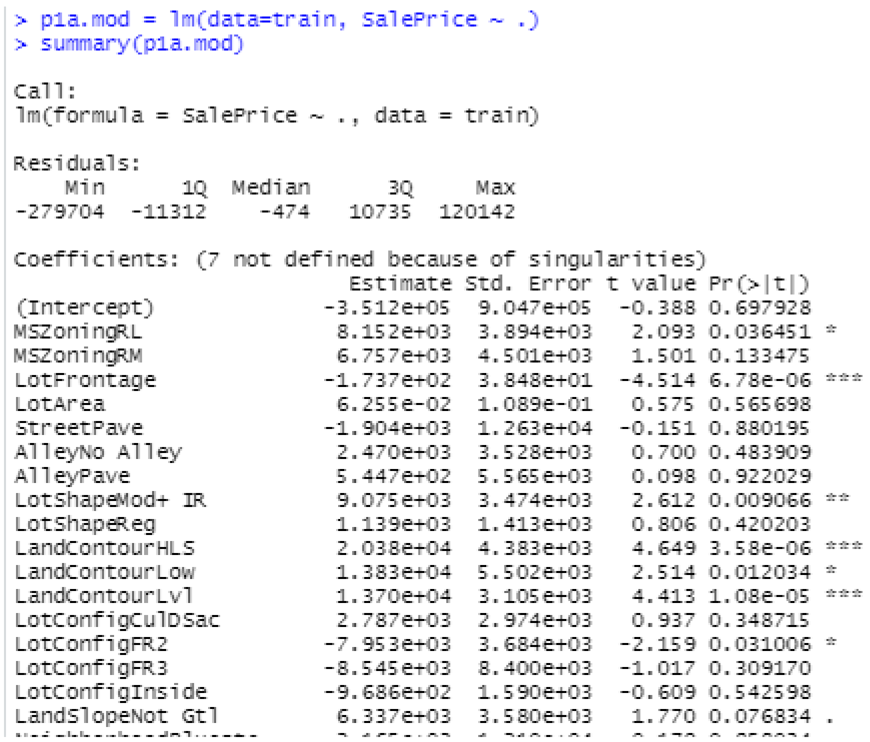
Tim Miller
15.071 | HW #3

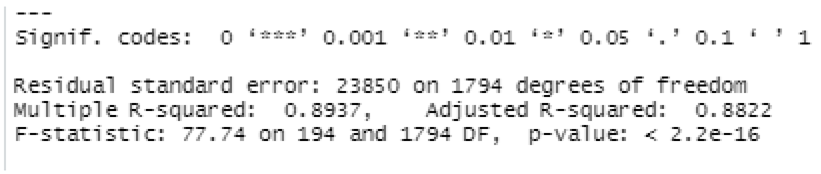
Problem 1a

Running a linear regression yields a model R^2 of 0.8937. See sample output below (not including everything because it is very long). Some coefficients are reported as NA because they overlap with other variables in the dataset so do not contribute to the model. For example TotalBsmtSF is just the sum of the other Basement Sq Ft metrics (finished / unfinished) so having TotalBsmtSF does not add anything.



```
> p1a.mod = lm(data=train, SalePrice ~ .)
> summary(p1a.mod)

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

Residuals:
    Min      1Q  Median      3Q     Max 
-279704  -11312    -474   10735  120142 

Coefficients: (7 not defined because of singularities)
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)             -3.512e+05  9.047e+05  -0.388 0.697928    
MSZoningRL               8.152e+03  3.894e+03   2.093 0.036451 *  
MSZoningRM               6.757e+03  4.501e+03   1.501 0.133475    
LotFrontage             -1.737e+02  3.848e+01  -4.514 6.78e-06 ***
LotArea                  6.255e-02  1.089e-01   0.575 0.565698    
StreetPave              -1.904e+03  1.263e+04  -0.151 0.880195    
AlleyNo Alley            2.470e+03  3.528e+03   0.700 0.483909    
AlleyPave                5.447e+02  5.565e+03   0.098 0.922029    
LotShapeMod+ IR          9.075e+03  3.474e+03   2.612 0.009066 ** 
LotShapeReg              1.139e+03  1.413e+03   0.806 0.420203    
LandContourHLS           2.038e+04  4.383e+03   4.649 3.58e-06 ***
LandContourLow           1.383e+04  5.502e+03   2.514 0.012034 *  
LandContourLvl           1.370e+04  3.105e+03   4.413 1.08e-05 ***
LotConfigCulDSac         2.787e+03  2.974e+03   0.937 0.348715    
LotConfigFR2            -7.953e+03  3.684e+03  -2.159 0.031006 *  
LotConfigFR3            -8.545e+03  8.400e+03  -1.017 0.309170    
LotConfigInside         -9.686e+02  1.590e+03  -0.609 0.542598    
LandSlopeNot Gtl         6.337e+03  3.580e+03   1.770 0.076834 .  
```

…
…
…
…



```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 23850 on 1794 degrees of freedom
Multiple R-squared:  0.8937,	Adjusted R-squared:  0.8822 
F-statistic: 77.74 on 194 and 1794 DF,  p-value: < 2.2e-16
```

Problem 1b

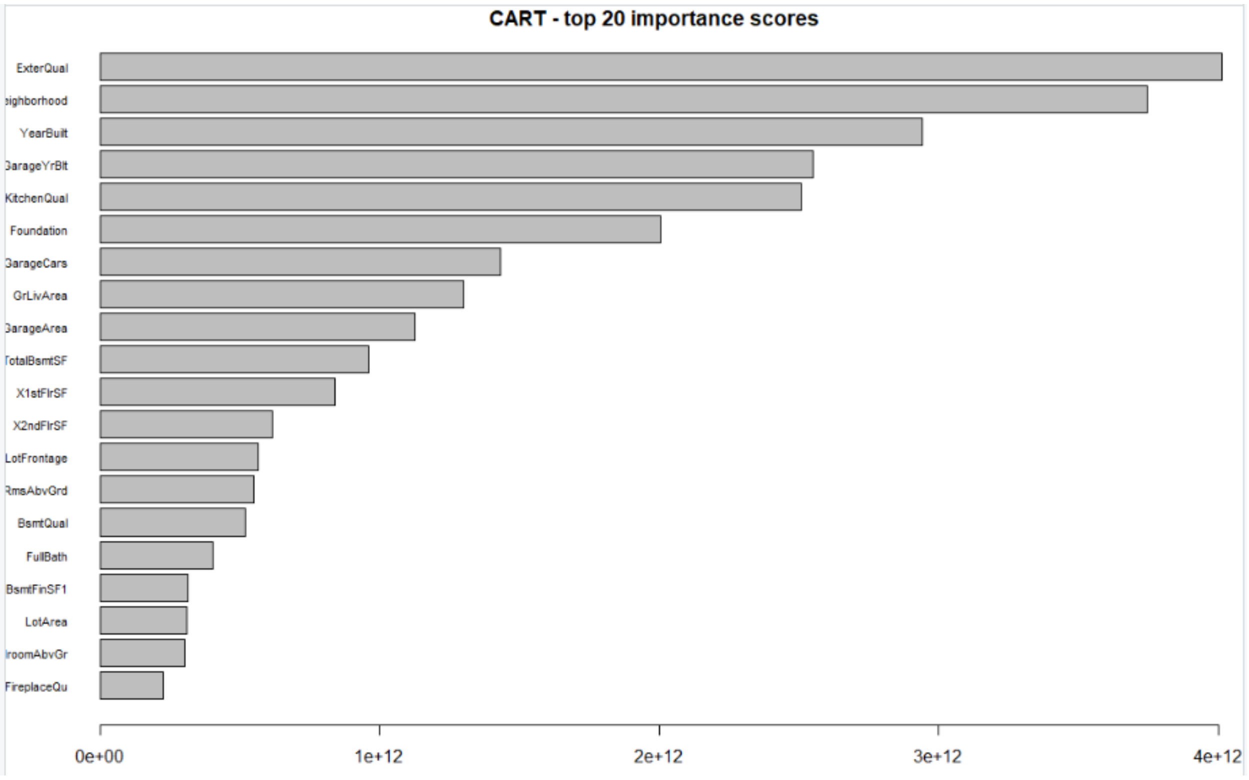
We want to find the cp value with the min RMSE. When we run the below code, we find the **cp value is 0.000296.**

```
> cv.results[which.min(cv.results$RMSE),]
          cp     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
287 0.000296 30913.59 0.8036813 21541.93 2275.525 0.02855522 1360.901
```

Code to train the model with above cp value:

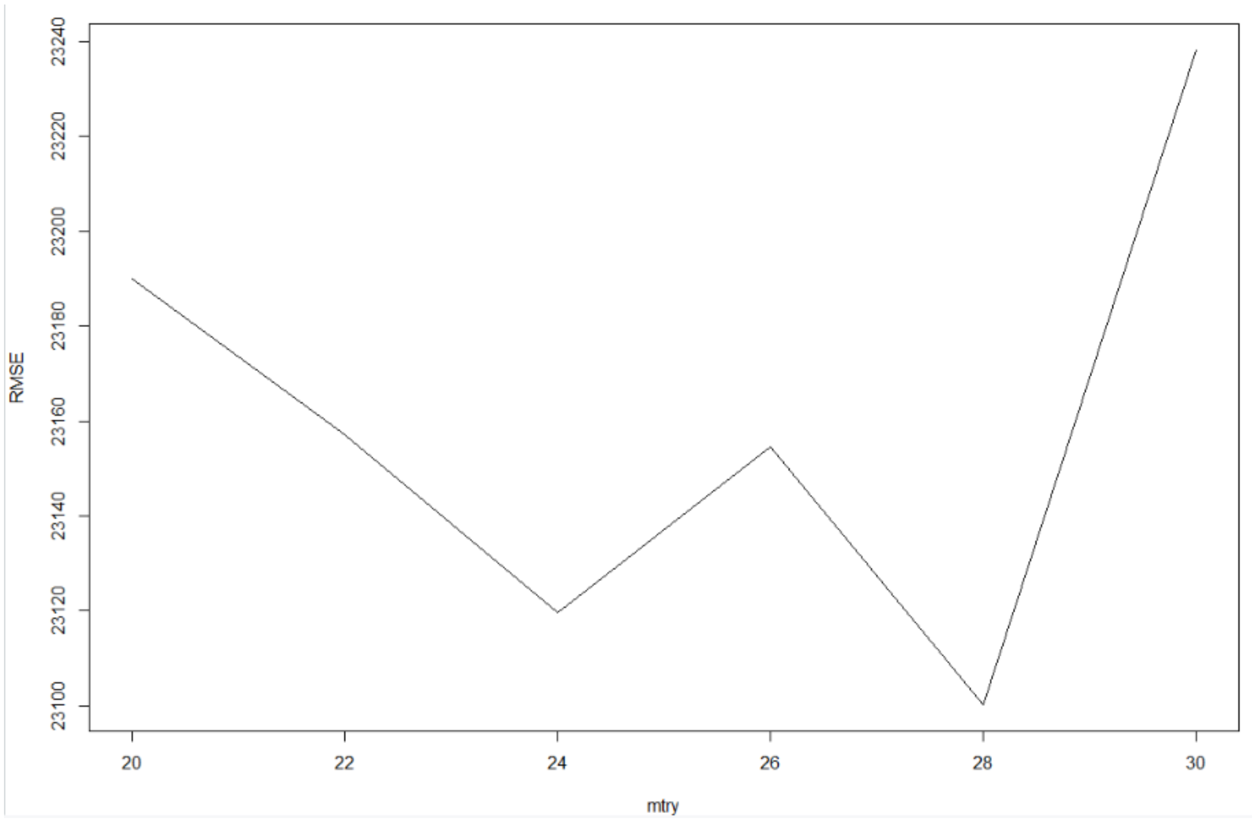
```
#train model
cart.mod = rpart(SalePrice ~ ., data = train, cp=0.000296)
prp(cart.mod)
```

Below we see the top 20 most important variables. Clear trend is that variables related to the quality and durability of the home are shown as important predictors for sale price (e.g., External Material Quality, Year Built, Foundation, Kitchen Quality). Perhaps Ames is known for some weather events and people want a sturdy house. Additionally, people probably want to buy a house that they don't need to fix up. I would imagine lower quality houses go for lower prices because the buyer and owner know money needs to go into the house to make improvements.



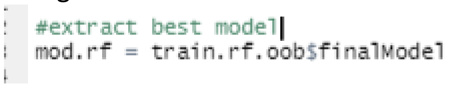
Problem 1c

Plotting RMSE vs mtry we find that that value of mtry is 28.



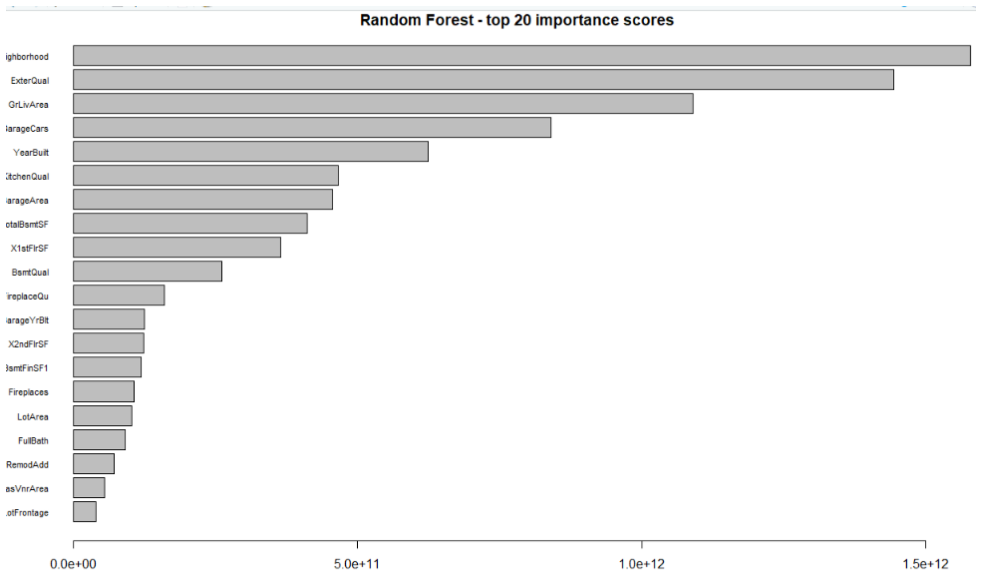
Problem 1d

Using the below code to extract the final model:



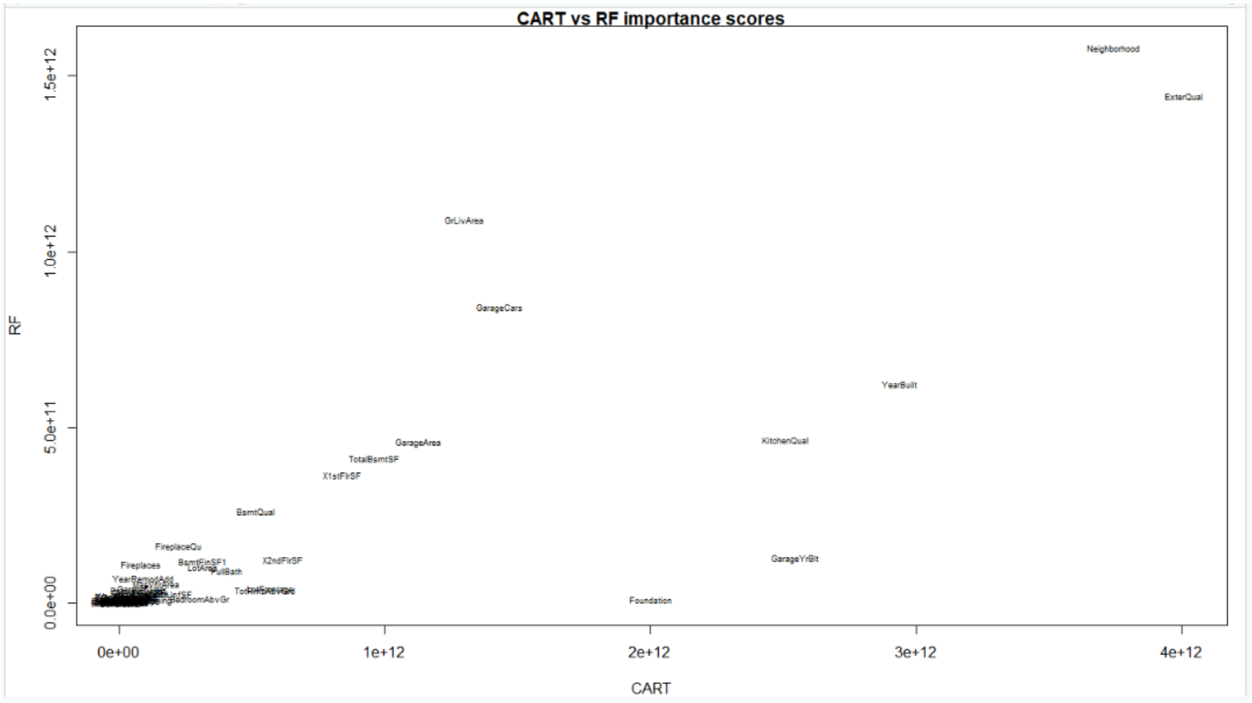
```
#extract best model
mod.rf = train.rf.oob$finalModel
```

We get the chart below for top 20 importance variables. The variables that appear in the Random Forest model seem well placed and intuitive. Most variables corresponding to the size and quality of the house.



Problem 1e

Interesting to see GrLivArea higher up in this model vs. the CART model. Intuitively I would have guessed that GrLivArea would be high up given people pay more money for bigger houses (i.e., surprised it was lower down in the CART model). Furthermore, foundation does not crack the top 20 for the Random Forest, whereas it is #4 important variable for CART. But that does not surprise me too much. I wonder how many perspective buyers *actually* look at the home foundation as they are making the bid on the home. Perhaps this is a random correlation picked up in the CART model that is corrected for in the Random Forest model.



Problem 1f

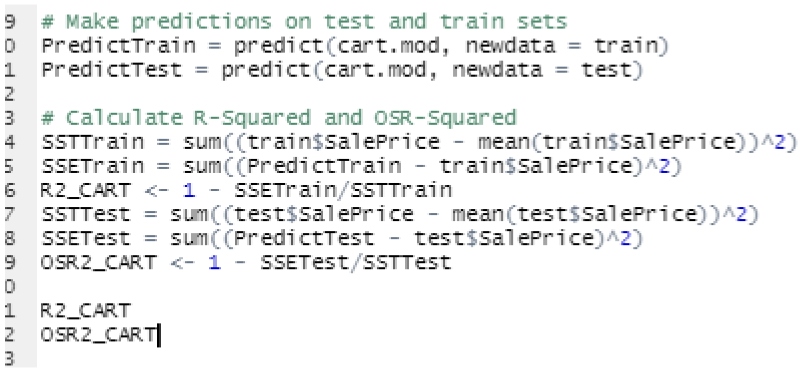
As expected the R^2 goes up from Linear Regression, to CART, to Random Forest. Also we can confirm that the OSR^2 for CART drops, but does not for Random Forest. A potential explanation here is that the CART model overfit to certain variables e.g., Foundation because they were more prevalent in the train set, but less in the test set. So when we run the model on the test set the predictive power is not as good because of the overfitting.

Linear regression

- In sample R^2 = 0.8937
- OSR^2 = 0.8854796

CART

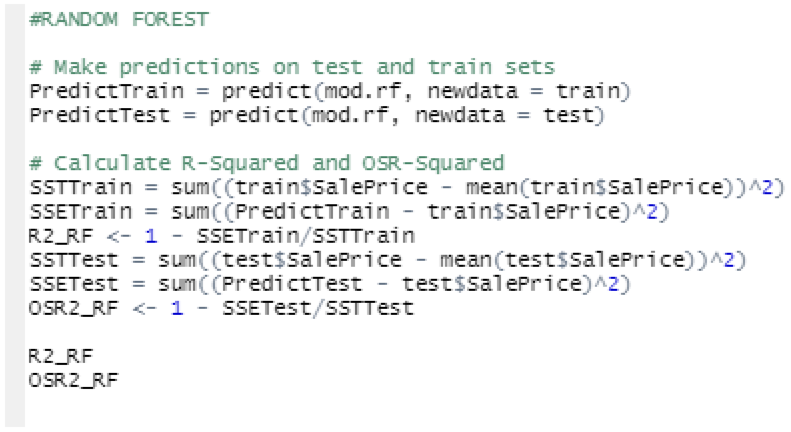
- In sample R^2 = 0.9108464
- OSR^2 = 0.7977558



```
# Make predictions on test and train sets
PredictTrain = predict(cart.mod, newdata = train)
PredictTest = predict(cart.mod, newdata = test)

# Calculate R-Squared and OSR-Squared
SSTTrain = sum((train$SalePrice - mean(train$SalePrice))^2)
SSETrain = sum((PredictTrain - train$SalePrice)^2)
R2_CART <- 1 - SSETrain/SSTTrain
SSTTest = sum((test$SalePrice - mean(test$SalePrice))^2)
SSETest = sum((PredictTest - test$SalePrice)^2)
OSR2_CART <- 1 - SSETest/SSTTest

R2_CART
OSR2_CART
```

Random Forest

- In sample R^2 = 0.9559223
- OSR^2 = 0.8851491



```
#RANDOM FOREST

# Make predictions on test and train sets
PredictTrain = predict(mod.rf, newdata = train)
PredictTest = predict(mod.rf, newdata = test)

# Calculate R-Squared and OSR-Squared
SSTTrain = sum((train$SalePrice - mean(train$SalePrice))^2)
SSETrain = sum((PredictTrain - train$SalePrice)^2)
R2_RF <- 1 - SSETrain/SSTTrain
SSTTest = sum((test$SalePrice - mean(test$SalePrice))^2)
SSETest = sum((PredictTest - test$SalePrice)^2)
OSR2_RF <- 1 - SSETest/SSTTest

R2_RF
OSR2_RF
```

Problem 1g

This would not be a fair comparison. Cross validation helps reduce some of the variability seen across training data. If we just trained the random forest on the test set we are not properly accounting for the randomness in the test data. Furthermore, the test data that we are using is a small sample – only 30% of the dataset. This small dataset further introduces a chance for randomness as we are training the random forest.

Problem 1h

i) As a property developer I would use the CART model. As we discussed, that model takes into account a number of different structural parts of the house to determine final price e.g., External Material, Foundation, Kitchen Quality, Theoretically the property developer would reason that investment in all these elements and increase the price of the house.

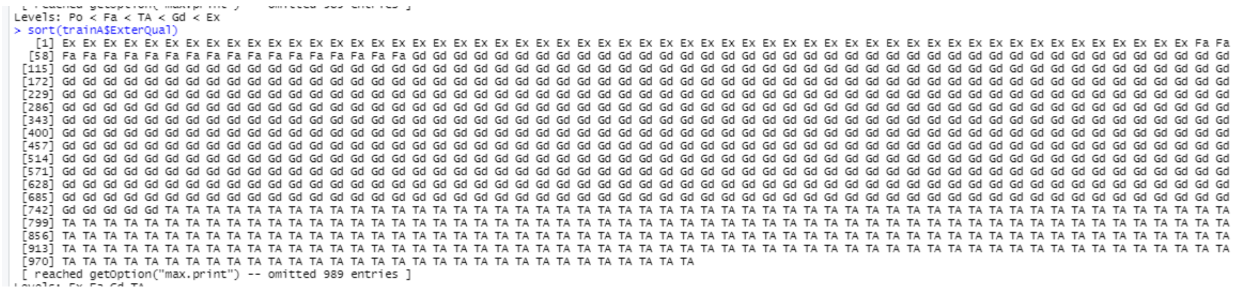
ii) As a real-estate firm, I would want to use the Random Forest Model because it has the best prediction power (highest in sample R^2 and 2nd highest OSR^2). This will allow us to advise our clients on the best listing and bid prices. The drawback here is that our clients would need to inherently trust a "black box" model. Likely they would not understand Random Forest models so would just have to trust in our ability to predict prices.

iii) As a value assessor, I would want to use the linear regression model. Although the model is not as accurate as the CART or Random Forest Models, it is much more intuitive to explain. If I have to defend the assessment in court proceedings, I want to have a clear and easy to explain model so that the judge or jury can understand my methodology. If I used CART or Random Forest, there are way too many branches on the trees that it would be really difficult to walk and audience through the model. The big limitation is that we have a lot of non-significant variables in the model. Ideally, we would want to remove those so it does not overwhelm the audience.

Problem 2a

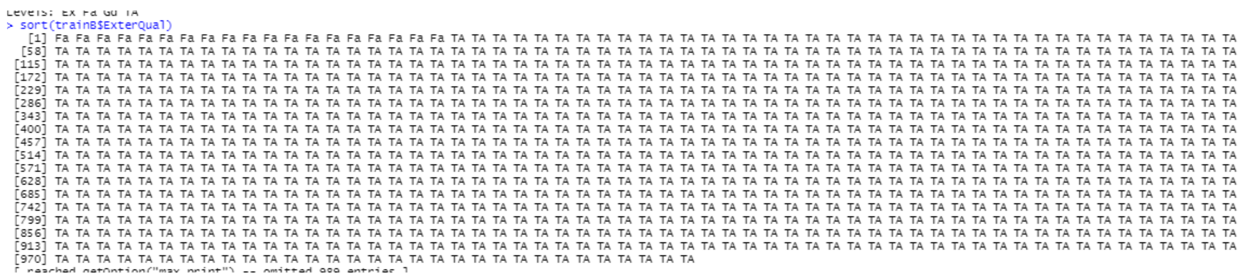
Sorting the ordered factors sorts based on the ordering rules that we have stated e.g., Po < Fa < TA < Gd < Ex. Sorting the unordered factors just sorts the data frame column.

For example, unordered sorting just orders alphabetically



```
Levels: Po < Fa < TA < Gd < Ex
> sort(trainA$ExterQual)
   [1] Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Ex Fa Fa
  [58] Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd
 [115] Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd
 [172] Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd
 [229] Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd
 [286] Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd
 [343] Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd
 [400] Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd
 [457] Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd
 [514] Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd
 [571] Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd
 [628] Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd
 [685] Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd Gd
 [742] Gd Gd Gd Gd Gd TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [799] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [856] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [913] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [970] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [ reached getOption("max.print") -- omitted 989 entries ]
```

Whereas ordered sorting sorts by the order we provided

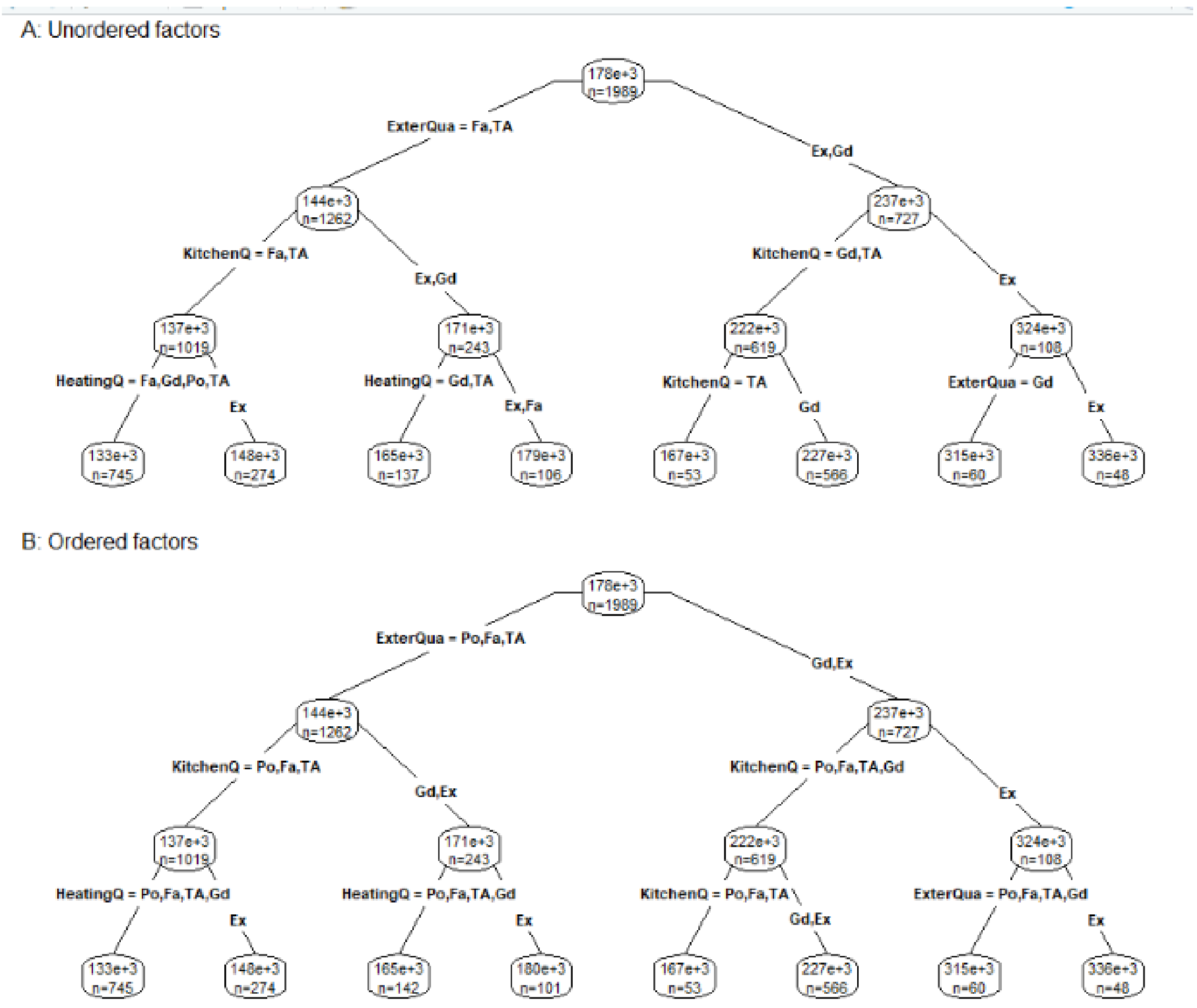


```
Levels: Ex Fa Gd TA
> sort(trainB$ExterQual)
   [1] Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa Fa TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
  [58] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [115] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [172] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [229] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [286] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [343] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [400] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [457] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [514] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [571] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [628] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [685] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [742] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [799] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [856] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [913] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [970] TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA TA
 [ reached getOption("max.print") -- omitted 989 entries ]
```

Problem 2b

We see that in the unordered factor tree there are some confusing splits. For example, on the bottom tree that is the second from the left Gd and TA are together and Ex,Fa are grouped together. Intuitively Gd and Ex should be grouped together and TA and FA should be grouped together because they are more similar in quality. Since we forced the order in the ordered tree, we do not run into this problem.

Furthermore, we see that the Po quality tag shows up much more often for the ordered tree than the unordered tree. This is evident in the first split where see Po show up for the ordered tree and not the unordered tree.



<u>Problem 2c</u>

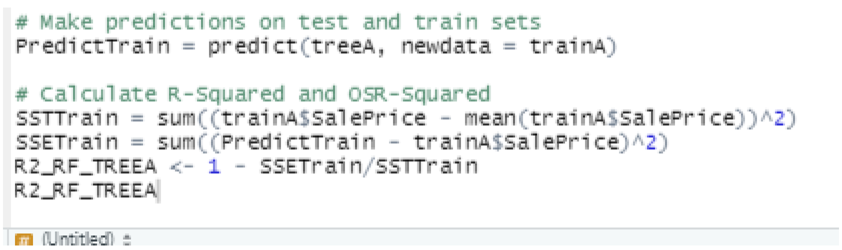
The ordered factor tree makes much more sense. As stated in 2b, we want to group the quality tags that are most similar together. Furthermore, we want to make sure that the Po quality tag is not lost in the analysis because it is at the end of the alphabet.

I would choose to use the ordered tree model in my analysis.

<u>Problem 2d</u>

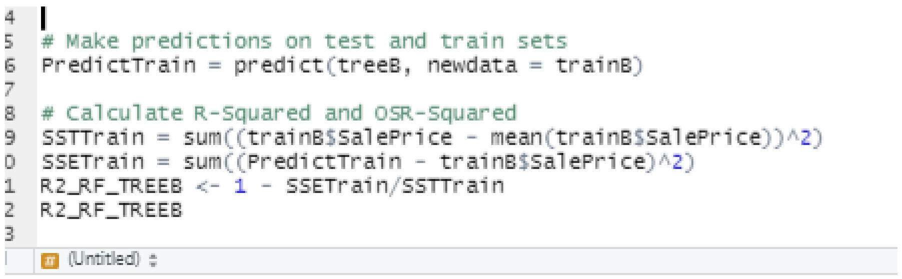
Based on R^2 it looks like TreeA fits the data better as it has a slightly higher R^2.

Tree A R^2 = 0.5641749



```
# Make predictions on test and train sets
PredictTrain = predict(treeA, newdata = trainA)

# Calculate R-Squared and OSR-Squared
SSTTrain = sum((trainA$SalePrice - mean(trainA$SalePrice))^2)
SSETrain = sum((PredictTrain - trainA$SalePrice)^2)
R2_RF_TREEA <- 1 - SSETrain/SSTTrain
R2_RF_TREEA
```

Tree B R^2 = 0.5641458

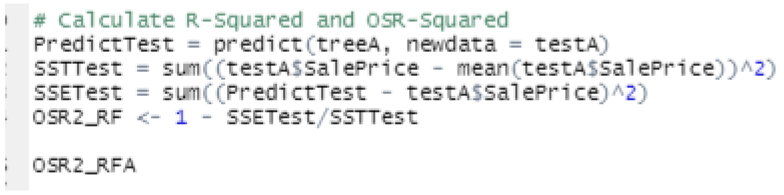


```

# Make predictions on test and train sets
PredictTrain = predict(treeB, newdata = trainB)

# Calculate R-Squared and OSR-Squared
SSTTrain = sum((trainB$SalePrice - mean(trainB$SalePrice))^2)
SSETrain = sum((PredictTrain - trainB$SalePrice)^2)
R2_RF_TREEB <- 1 - SSETrain/SSTTrain
R2_RF_TREEB
```

Problem 2e

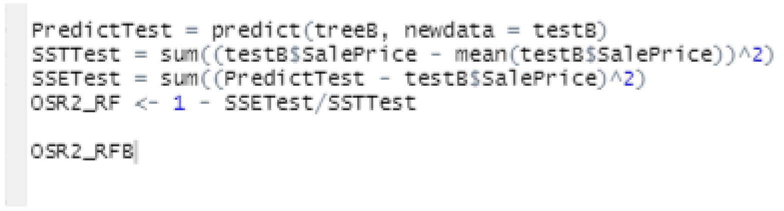
It looks like Tree B performs better on the test set as it has a slightly higher OSR^2.

Tree A OSR^2 = 0.5337154



```
# Calculate R-Squared and OSR-Squared
PredictTest = predict(treeA, newdata = testA)
SSTTest = sum((testA$SalePrice - mean(testA$SalePrice))^2)
SSETest = sum((PredictTest - testA$SalePrice)^2)
OSR2_RF <- 1 - SSETest/SSTTest

OSR2_RFA
```

Tree B OSR^2 = 0.5358374



```
PredictTest = predict(treeB, newdata = testB)
SSTTest = sum((testB$SalePrice - mean(testB$SalePrice))^2)
SSETest = sum((PredictTest - testB$SalePrice)^2)
OSR2_RF <- 1 - SSETest/SSTTest

OSR2_RFB
```

Problem 2f

Ordered factoring reduces the run time for CART models. By introducing logical ordering, we reduce the search time of the algorithm to find the best split. For example, with the models we developed in problem 2. With the ordered values, we help group the similar quality levels close to each other. Intuitively we know that Po and Fa should be grouped close together and we get a better model *when* they are grouped together. By doing this for the computer, it does not have to go out and try additional combinations to discover that through trial and error.

<u>Problem 2g</u>

An example that might work is gradient of hair color as a predictor of standardized test scores.

Hair color gradient has clear and logical ordering: you would want to order from dark to light (or vice versa).

But, intuitively, I don't see a scenario where hair color matters at all for predicting standardized test scores. Therefore we would violate the second condition.