In [34]:

```
# Load in data exclude 2 variabes
library(tidyverse)


ameslist <- read.table("https://msudataanalytics.github.io/SSC442/Labs/data/ames.csv
                       header = TRUE,
                       sep = ",")
df <- (ameslist)
ames = subset(df, select = -c(OverallCond,OverallQual))
```

In [35]:

```
#create series of models for prediction of RMSE

library(MASS)

model1 <- lm(SalePrice ~1,data = ames)

model2 <- lm(SalePrice ~ (LotArea), data = ames)

model3 <- lm(SalePrice ~ (LotArea+  YearBuilt), data = ames)

model4 <- lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF), data = ames)

model5 <- lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF), data = ames)

model6 <- lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                            BsmtFullBath), data = ames)

model7 <- lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                            BsmtFullBath+ BsmtHalfBath), data = ames)

model8 <- lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                            BsmtFullBath+ BsmtHalfBath+BedroomAbvGr), data = ames)

model9 <- lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                            BsmtFullBath+ BsmtHalfBath+BedroomAbvGr+KitchenAbvGr), d

model10 <- lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                             BsmtFullBath+ BsmtHalfBath+BedroomAbvGr+KitchenAbvGr+
                             TotRmsAbvGrd), data = ames)

model11 <- lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                             BsmtFullBath+ BsmtHalfBath+BedroomAbvGr+KitchenAbvGr+
                             TotRmsAbvGrd+ GarageCars), data = ames)

model12 <- lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                             BsmtFullBath+ BsmtHalfBath+BedroomAbvGr+KitchenAbvGr+
                             TotRmsAbvGrd+ GarageCars+WoodDeckSF), data = ames)
```

```r
model13 <- lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                             BsmtFullBath+ BsmtHalfBath+BedroomAbvGr+KitchenAbvGr+
                             TotRmsAbvGrd+ GarageCars+WoodDeckSF+ScreenPorch), data

model14 <- lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                             BsmtFullBath+ BsmtHalfBath+BedroomAbvGr+KitchenAbvGr+
                             TotRmsAbvGrd+ GarageCars+WoodDeckSF+ScreenPorch+
                             PoolArea), data = ames)

model15 <- lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                             BsmtFullBath+ BsmtHalfBath+BedroomAbvGr+KitchenAbvGr+
                             TotRmsAbvGrd+ GarageCars+WoodDeckSF+ScreenPorch+
                             PoolArea+CentralAir), data = ames)

model16 <- lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                             BsmtFullBath+ BsmtHalfBath+BedroomAbvGr+KitchenAbvGr+
                             TotRmsAbvGrd+ GarageCars+WoodDeckSF+ScreenPorch+
                             PoolArea+MSSubClass+CentralAir), data = ames)
```

In [36]:

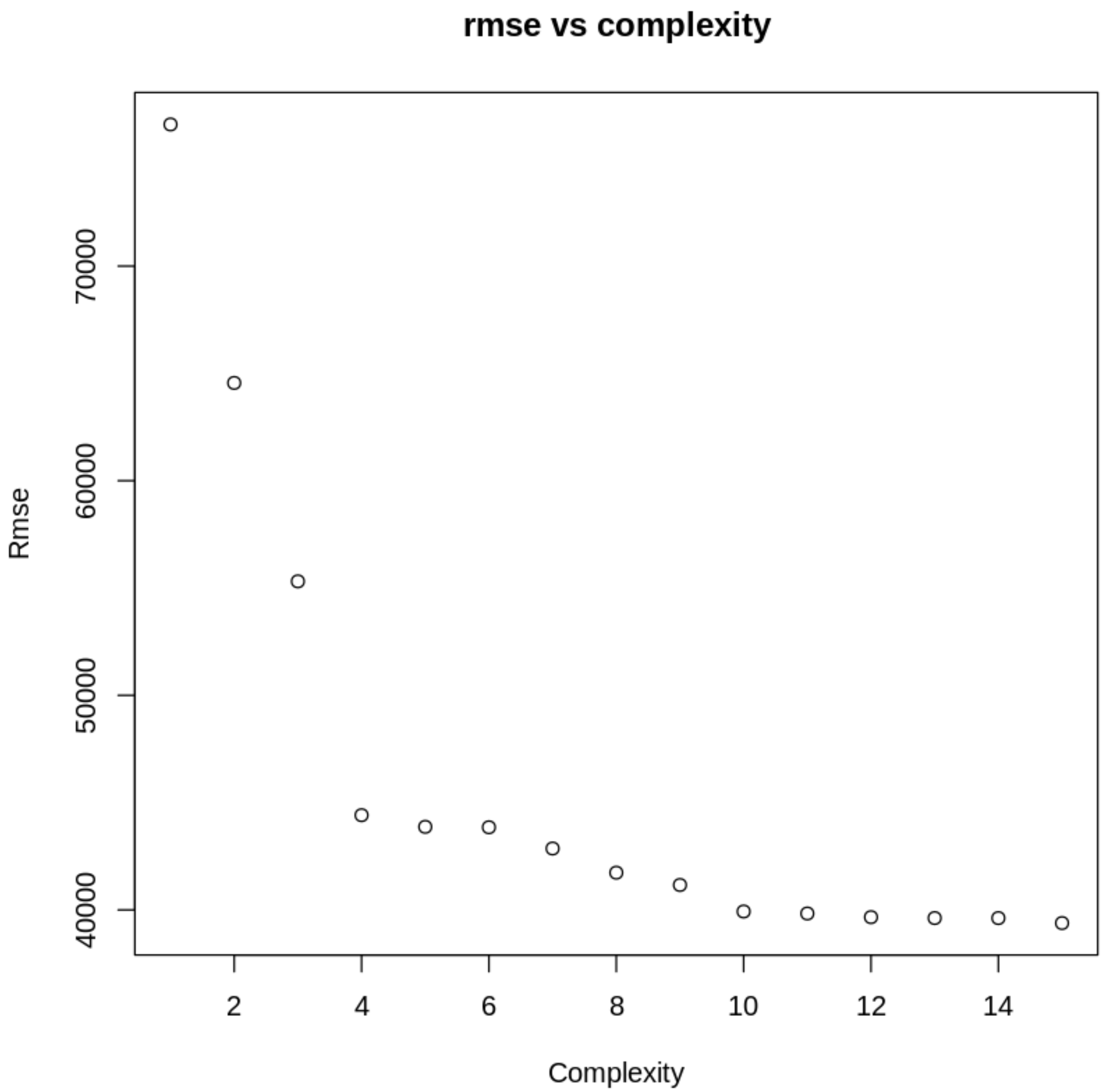
```r
test <- lm(SalePrice~GarageCars, data=ames)
```

In [37]:

```r
get_complexity = function(model) {

  length(coef(model)) - 1

}

#given rmse

rmse = function(actual, predicted) {

  sqrt(mean((actual - predicted) ^ 2))

}
```

In [38]:

```
chart <- plot(c(get_complexity(model2),get_complexity(model3)
                ,get_complexity(model4),get_complexity(model5),get_complexity(model6
                get_complexity(model7),get_complexity(model8),get_complexity(model9)
                ,get_complexity(model11),get_complexity(model12),get_complexity(mode
                ,get_complexity(model15),get_complexity(model16)),c(rmse(ames$SalePr
                                                                    rmse(ames$SalePr
                                                                    rmse(ames$SalePr
                                                                    rmse(ames$SalePr
                                                                    rmse(ames$SalePr
                                                                    rmse(ames$SalePr
                                                                    rmse(ames$SalePr
                                                                    rmse(ames$SalePr
              xlab="Complexity", ylab="Rmse")
```



Describe any patterns you see.

Do you think you should use the full-size model? Why or why not? What criterion are you using to make this statement?

- **The higher the complexity, the lower the rmse. This means that a model with greater complexity is likely to give is better results, but we must be careful not to overfit at the same time. Using a full model leads to more risk of overfitting, so striking that balance is important. It is more important to find data that correlates well with your target variable than it is to have a large, complex model.**

In [39]:

```
set.seed(9)
num_obs = nrow(ames)

train_index = sample(num_obs, size = trunc(0.50 * num_obs))
train_data = ames[train_index, ]
test_data = ames[-train_index, ]
```

In [40]:

```
fit_0 = lm(SalePrice ~ 1, data = train_data)
fit_1=lm(SalePrice ~ (LotArea), data = ames)
fit_2=lm(SalePrice ~ (LotArea+  YearBuilt), data = ames)
fit_3=lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF), data = ames)
fit_4=lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF), data = ames)
fit_5=lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                        BsmtFullBath), data = ames)
fit_6=lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                        BsmtFullBath+ BsmtHalfBath), data = ames)
fit_7=lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                        BsmtFullBath+ BsmtHalfBath), data = ames)
fit_8=lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                        BsmtFullBath+ BsmtHalfBath+BedroomAbvGr), data = ames)
fit_9=lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                        BsmtFullBath+ BsmtHalfBath+BedroomAbvGr+KitchenAbvGr), data
fit_10=lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                         BsmtFullBath+ BsmtHalfBath+BedroomAbvGr+KitchenAbvGr+
                         TotRmsAbvGrd), data = ames)
fit_11=lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                         BsmtFullBath+ BsmtHalfBath+BedroomAbvGr+KitchenAbvGr+
                         TotRmsAbvGrd+ GarageCars), data = ames)
fit_12=lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                         BsmtFullBath+ BsmtHalfBath+BedroomAbvGr+KitchenAbvGr+
                         TotRmsAbvGrd+ GarageCars+WoodDeckSF), data = ames)
fit_13=lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                         BsmtFullBath+ BsmtHalfBath+BedroomAbvGr+KitchenAbvGr+
                         TotRmsAbvGrd+ GarageCars+WoodDeckSF+ScreenPorch), data = an
fit_14=lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                         BsmtFullBath+ BsmtHalfBath+BedroomAbvGr+KitchenAbvGr+
                         TotRmsAbvGrd+ GarageCars+WoodDeckSF+ScreenPorch+
                         PoolArea), data = ames)
fit_15=lm(SalePrice ~ (LotArea+  YearBuilt+X1stFlrSF+X2ndFlrSF+
                         BsmtFullBath+ BsmtHalfBath+BedroomAbvGr+KitchenAbvGr+
                         TotRmsAbvGrd+ GarageCars+WoodDeckSF+ScreenPorch+
                         PoolArea+MSSubClass), data = ames)
get_complexity(fit_1)
```

1

In [41]:

```
# train RMSE

print(paste0("Train: ", sqrt(mean((train_data$SalePrice - predict(fit_0, train_

# test RMSE
print(paste0("Test: ", sqrt(mean((test_data$SalePrice - predict(fit_0, test_data

# train RMSE
print(paste0("Train: ", rmse(actual = train_data$SalePrice, predicted = predict
# test RMSE
print(paste0("Test: ", rmse(actual = test_data$SalePrice, predicted = predict(fi
```

```
[1] "Train: 80875.9784504071"
[1] "Test: 77928.6200203521"
[1] "Train: 80875.9784504071"
[1] "Test: 77928.6200203521"
```

In [52]:

```
get_rmse = function(model, data, response) {
  rmse(actual = subset(data, select = response, drop = TRUE),
       predicted = predict(model, data))
}

print(paste0("Output: ", get_rmse(model = fit_0, data = train_data, response = "Sale
print(paste0("Output: ", get_rmse(model = fit_0, data = test_data, response = "SaleP
```

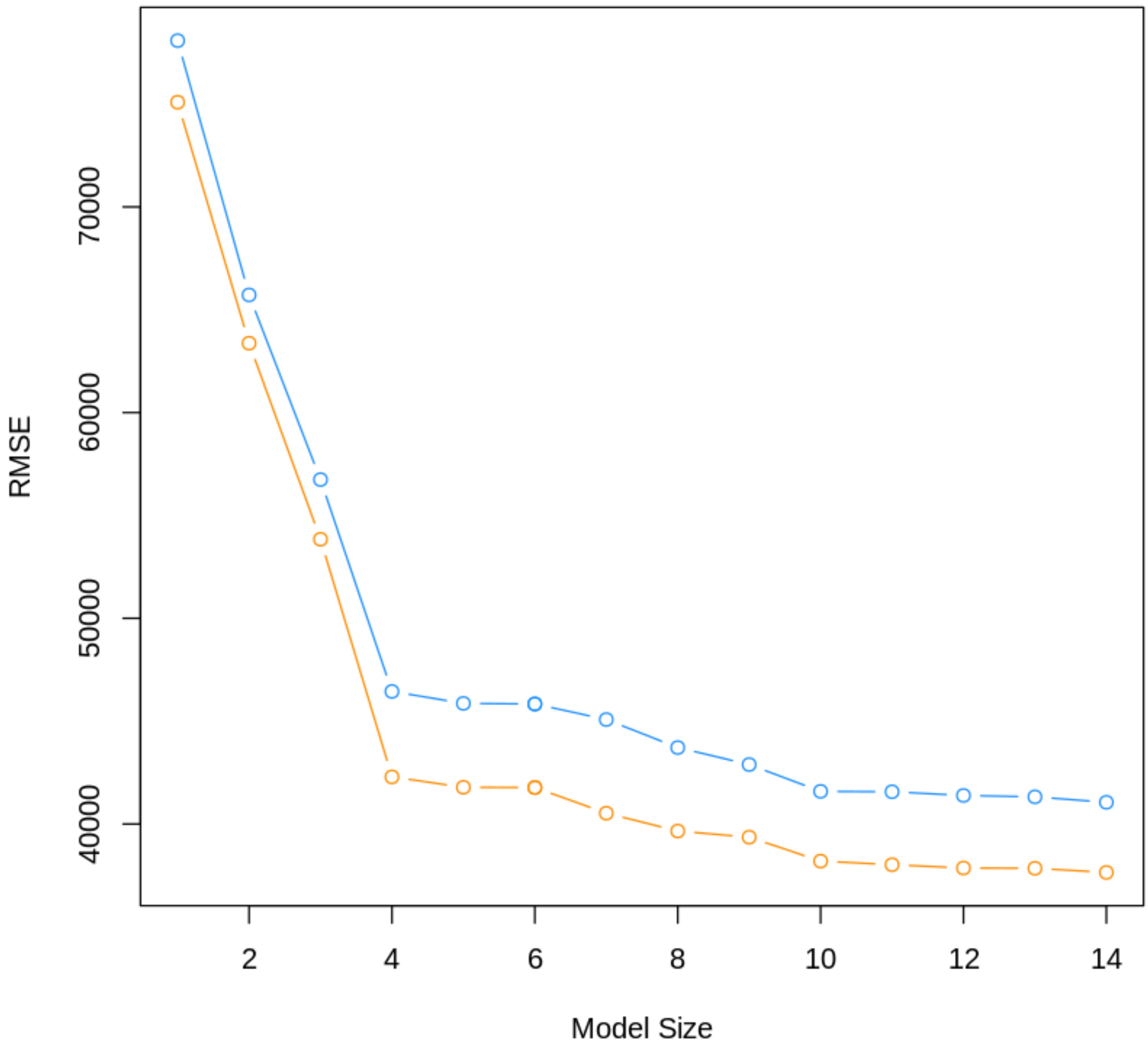
```
[1] "Output: 80875.9784504071"
[1] "Output: 77928.6200203521"
```

In [53]:

```
model_list = list(fit_1, fit_2, fit_3, fit_4, fit_5,fit_6,fit_7,fit_8,fit_9,fit_10,f
                  ,fit_13,fit_14,fit_15)


train_rmse = sapply(model_list, get_rmse, data = train_data, response = "SalePrice")
test_rmse = sapply(model_list, get_rmse, data = test_data, response = "SalePrice")
model_complexity = sapply(model_list, get_complexity)
```

In [54]:

```
plot(model_complexity, train_rmse, type = "b",
     ylim = c(min(c(train_rmse, test_rmse)) - 0.02,
              max(c(train_rmse, test_rmse)) + 0.02),
     col = "dodgerblue",
     xlab = "Model Size",
     ylab = "RMSE")
lines(model_complexity, test_rmse, type = "b", col = "darkorange")
```



Final Model

In [62]:

```
finalmodel <- lm(SalePrice ~ (LotArea+YearBuilt+X1stFlrSF+X2ndFlrSF+
        BedroomAbvGr+KitchenAbvGr+TotRmsAbvGrd+GarageCars+ScreenPorch), data = ames)
```

In [63]:

```
cat("Train RMSE: ",get_rmse(model = finalmodel, data = train_data, response = "Sale
```

```
Train RMSE:  42013.93
```

In [64]:

```
cat("Test RMSE: ",get_rmse(model = finalmodel, data = test_data, response = "SalePr
```

```
Test RMSE:  38458.69
```

In [72]:

```
#Exercise 2 part 2

set.seed(9)
num_obs = nrow(ames)

train_index = sample(num_obs, size = trunc(0.50 * num_obs))
train_data = ames[train_index, ]
test_data = ames[-train_index, ]
get_complexity(fit_0)

# train RMSE
sqrt(mean((train_data$SalePrice - predict(finalmodel, train_data)) ^ 2))
# test RMSE
sqrt(mean((test_data$SalePrice - predict(finalmodel, test_data)) ^ 2))

# train RMSE
cat("Actual Train: ",rmse(actual = train_data$SalePrice, predicted = predict(fi
    "Actual Test: ",rmse(actual = test_data$SalePrice, predicted = predict(fina
```

0

42013.9294811871

38458.6942772405

```
Actual Train:  42013.93 Actual Test:  38458.69
```

In [57]:

```
get_rmse = function(model, data, response) {
  rmse(actual = subset(data, select = response, drop = TRUE),
       predicted = predict(model, data))
}
```

Final Explanation

For our final model, we came to the conclusion that the more variables we had in the model, The greater chsance of overfitting there was. To this end, we looked at the previous calculations of rmse and decided that 10 variables seems to be the maximum number before the rmse starts to experience dimishing returns to its reduction. Our final model is comprised of at maximum 10 predictors. We decided not to use interaction terms in order to communicate more clearly which variables predict sale price most strongly on their own. These were the variables that correlated most closely with sale price when ran through linear regression.
This is the most concrete conclusion we have come to so far, but we know there are edits we can make to our code and model selection that would most likely cut down the rmse further.

In [ ]:

```
```