Lab No. 8

Elizabeth Walter

```
knitr::opts_chunk$set(echo = TRUE,
                      fig.align = 'center')
```

```
library(skimr)
library(ggplot2)
Ames2 <- read.table("https://ssc442.netlify.app/projects/data/ames.csv",
                 header = TRUE,
                 sep = ",")
```

1. Load the Ames data. Using skimr::skim, find the variables that have a complete rate of below 60% and drop them.

Dropped: Alley (completion rate = 0.0623), FireplaceQu (c.r. = 0.527), PoolQC (0.00479), Fence (0.192), MiscFeature (0.0370)

Note: I removed 'skimr::skim(Ames2)' from the code chunk because I could not get this to knit with the output of it displaying.

```
Ames2 <- subset(Ames2, select = -c(Alley, FireplaceQu, PoolQC, Fence, MiscFeature))
```

2. Take a look at Utilities. Use the table function to see a tabulation of the values of Utilities. Do you see why this field is not likely to be useful to us, or even problematic?

This is problematic because all but one of its observations are of the same value, which provides almost no variance or insight.

```
table(Ames2$Utilities)
```

```
##
## AllPub NoSeWa
##   1459      1
```

3. Using forward selection (that is, select one variable, then select another) create a series of models up to complexity length 15. You may use any variable within the dataset, including categorical variables.

```
fit_1 <- lm(SalePrice ~ OverallQual, data = Ames2)
fit_2 <- lm(SalePrice ~ OverallQual + YearBuilt, data = Ames2)
fit_3 <- lm(SalePrice ~ OverallQual + YearBuilt + GrLivArea, data = Ames2)
fit_4 <- lm(SalePrice ~ OverallQual + YearBuilt + GrLivArea + OverallQual:YearBuilt,
            data = Ames2)
fit_5 <- lm(SalePrice ~ OverallQual + YearBuilt + GrLivArea + OverallQual:YearBuilt +
              TotRmsAbvGrd, data = Ames2)
fit_6 <- lm(SalePrice ~ OverallQual + YearBuilt + GrLivArea + OverallQual:YearBuilt +
              TotRmsAbvGrd + GarageCars, data = Ames2)
fit_7 <- lm(SalePrice ~ OverallQual + YearBuilt + GrLivArea + OverallQual:YearBuilt +
              TotRmsAbvGrd + GarageCars + BedroomAbvGr, data = Ames2)
fit_8 <- lm(SalePrice ~ OverallQual + YearBuilt + GrLivArea + OverallQual:YearBuilt +
              TotRmsAbvGrd + GarageCars + BedroomAbvGr + BldgType, data = Ames2)
fit_9 <- lm(SalePrice ~ OverallQual + YearBuilt + GrLivArea + OverallQual:YearBuilt +
              TotRmsAbvGrd + GarageCars + BedroomAbvGr + BldgType + Street, data = Ames2)
fit_10 <- lm(SalePrice ~ OverallQual + YearBuilt + GrLivArea + OverallQual:YearBuilt +
               TotRmsAbvGrd + GarageCars + BedroomAbvGr + BldgType + Street +
               TotalBsmtSF:GrLivArea, data = Ames2)
fit_11 <- lm(SalePrice ~ OverallQual + YearBuilt + GrLivArea + OverallQual:YearBuilt +
               TotRmsAbvGrd + GarageCars + BedroomAbvGr + BldgType + Street +
               TotalBsmtSF:GrLivArea + YrSold, data = Ames2)
fit_12 <- lm(SalePrice ~ OverallQual + YearBuilt + GrLivArea + OverallQual:YearBuilt +
               TotRmsAbvGrd + GarageCars + BedroomAbvGr + BldgType + Street +
               TotalBsmtSF:GrLivArea + YrSold + HouseStyle, data = Ames2)
fit_13 <- lm(SalePrice ~ OverallQual + YearBuilt + GrLivArea + OverallQual:YearBuilt +
               TotRmsAbvGrd + GarageCars + BedroomAbvGr + BldgType + Street +
               TotalBsmtSF:GrLivArea + YrSold + HouseStyle + LotArea, data = Ames2)
fit_14 <- lm(SalePrice ~ OverallQual + YearBuilt + GrLivArea + OverallQual:YearBuilt +
               TotRmsAbvGrd + GarageCars + BedroomAbvGr + BldgType + Street +
               TotalBsmtSF:GrLivArea + YrSold + HouseStyle + LotArea + CentralAir,
             data = Ames2)
fit_15 <- lm(SalePrice ~ OverallQual + YearBuilt + GrLivArea + OverallQual:YearBuilt +
               TotRmsAbvGrd + GarageCars + BedroomAbvGr + BldgType + Street +
               TotalBsmtSF:GrLivArea + YrSold + HouseStyle + LotArea + CentralAir +
               GarageArea, data = Ames2)
```

4. Create a chart plotting the model complexity as the x-axis variable and RMSE as the y-axis variable. 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?

Based on this I would think that the best model to use is the full model minus 1 variable because overall RMSE continues to decrease as model complexity increases until the last two models. There is a sharp decline in RMSE from the second to the third variable. From model complexity = 3 onward, there is some fluctuation in change, but there is a general pattern of decline. The large dashes between dots that are longer than the distance of a one variable model increase is from the dummy variables created by the categorical variables I included, which is why the model complexity extends beyond 15. Some variables have greater effect on the change in RMSE than others, and one or two even look to very slightly increase RMSE. The differences in effects of the variables on RMSE also makes me think that if I removed select variables, (not in the order that I added them to the model when doing forward selection, but the ones that have appear to have little effect) I could potentially get a less complex model that still significantly reduces RMSE. The less complex model may be better for predictions.

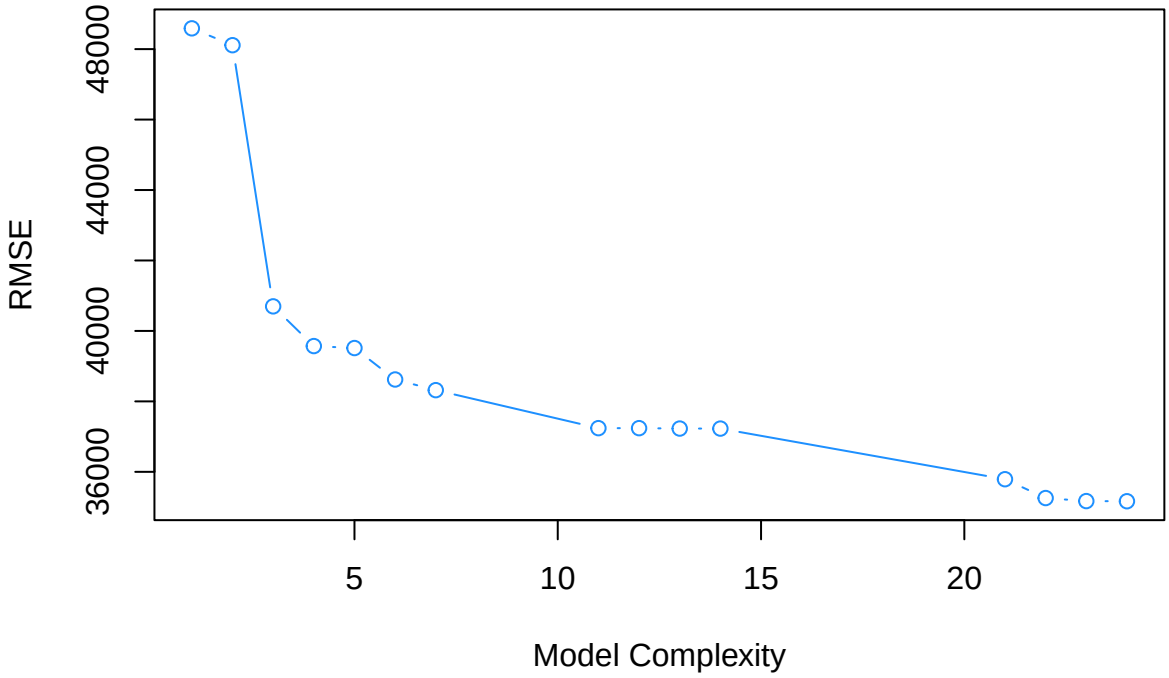
```
rmse = function(actual, predicted) {
  sqrt(mean((actual - predicted) ^ 2))
}

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

get_complexity <- function(model) {
  length(coef(model)) - 1
}
model_list <- list(fit_1, fit_2, fit_3, fit_4, fit_5, fit_6, fit_7, fit_8, fit_9,
                   fit_10, fit_11, fit_12, fit_13, fit_14, fit_15)

rmse <- sapply(model_list, get_rmse, data = Ames2, response = "SalePrice")
model_complexity <- sapply(model_list, get_complexity)

plot(model_complexity, rmse, type = "b",
     ylim = c(min(rmse) - 0.02,
              max(rmse) + 0.02),
     col = "dodgerblue",
     xlab = "Model Complexity",
     ylab = "RMSE")
```


RMSE
48000
44000
40000
36000
5
10
15
20
Model Complexity