Data Wrangling Capstone

## Data

The data was taken from <https://www.kaggle.com/c/house-prices-advanced-regression-techniques>. This project is to use the information provided and see how accurately I can predict the sales price of a house. Because it is designed for a competition, there isn’t as much data wrangling needed as a standard dataset that would have missing values, incomplete information, misinformation, etc.

1. After loading the dataset and using View() to check it out, I made sure to go through and understand what the different variables were and how they may correspond with each other.
2. My mentor and I decided that having as many factor variables as I did, it would be easy to convert those using dummy code into series of dichotomous variables (0 and 1) for each level.

* To do this I found the variables that were of class factor and assigned them to categFactors, which I then used in a for loop with dummy code from the psych library to convert any variables with those names and create a new matrix, while also assigning appropriate names based on the factor levels from each original variable, as seen here:

# Create a variable with all the names of columns which are factors, and one without  
classArray <- unlist(lapply(trainData, class))  
categFactors <- names(trainData)[which(classArray == "factor")]  
  
# Apply dummy coding to all categorical variables and create new variables at end of df  
dummyData = matrix(0, nrow(trainData), 1)  
for(i in 1:ncol(trainData[, categFactors])){ #1:ncol = 43  
 tempDummy <- dummy.code(trainData[, categFactors][, i])  
 colnames(tempDummy) <- paste(categFactors[i], colnames(tempDummy))  
 dummyData <- cbind(dummyData, tempDummy) #binding w/o col  
}

1. I then removed all the original factor variables and merged my factorless dataset with the newly created dummy coded set, while removing the blank variable I made up and filled with 0’s so as to not mess up my predictive modelling down the road.

#Removing all original factor columns   
categNotfactor <- names(trainData)[which(unlist(lapply(trainData, class)) != "factor")]  
  
#Combine factorless trainData with dummy code  
trainData <- cbind(trainData[, categNotfactor], dummyData)  
  
#remove dummy row of all 0's created earlier for cbind  
trainData <- trainData[, -39]

1. I then decided to look for any NA’s, NAN’s and NULL in my data to handle them as seen fit

* This helped narrow down where the NA’s were and led me to figure out how to handle them.

1. Using kNNimpute technique, I was able to take the variables discovered above and fill in the missing values as best predicted.

My data was now ready for me to take to the next step and begin modelling and predicting.