Annotated Appendix \_ EDA Final Project Report

Tam Nguyen

October 28th, 2018

### Exploring missing value

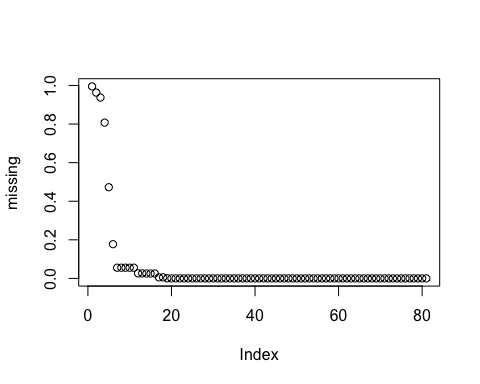
nrows <- nrow(train) #Numer of rows in train dataset  
missing <- sort(map\_dbl(train, function(x) sum(is.na(x)) / nrows), decreasing = TRUE)  
names\_missing <- names(missing[missing > 0])  
head(missing, 20)

## PoolQC MiscFeature Alley Fence FireplaceQu   
## 0.9952054795 0.9630136986 0.9376712329 0.8075342466 0.4726027397   
## LotFrontage GarageType GarageYrBlt GarageFinish GarageQual   
## 0.1773972603 0.0554794521 0.0554794521 0.0554794521 0.0554794521   
## GarageCond BsmtExposure BsmtFinType2 BsmtQual BsmtCond   
## 0.0554794521 0.0260273973 0.0260273973 0.0253424658 0.0253424658   
## BsmtFinType1 MasVnrType MasVnrArea Electrical Id   
## 0.0253424658 0.0054794521 0.0054794521 0.0006849315 0.0000000000

summary(missing)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.0589 0.0000 0.9952

plot(missing)



* Nineteen (19) of the original 80 variables have some degree of missing values. To begin addressing this problem we work on x\_data for this analysis.

#Make an copy of train dataset to work on 19 variables seprately   
x\_data <- train  
names\_missing\_del <- names(missing[missing > 0.8])  
x\_data <- select(x\_data, one\_of(setdiff(names(x\_data),names\_missing\_del ))) #new training dataset without 4 highest missing predictors

sum(is.na(train)) #how many data is missing

## [1] 6965

sum(is.na(train))/(1460\*80) #missing value percentage

## [1] 0.05963185

### Continuous predictor

For the initial iteration of the problem we first focus on those continuous predictor values. An investigation will be carried out to find good performing models with a focus on identifying (if any) the gap between simple explainable models and the more complex predictive models.

num\_data <- select\_if(x\_data, is.numeric);   
summary(num\_data)

## Id MSSubClass LotFrontage LotArea   
## Min. : 1.0 Min. : 20.0 Min. : 21.00 Min. : 1300   
## 1st Qu.: 365.8 1st Qu.: 20.0 1st Qu.: 59.00 1st Qu.: 7554   
## Median : 730.5 Median : 50.0 Median : 69.00 Median : 9478   
## Mean : 730.5 Mean : 56.9 Mean : 70.05 Mean : 10517   
## 3rd Qu.:1095.2 3rd Qu.: 70.0 3rd Qu.: 80.00 3rd Qu.: 11602   
## Max. :1460.0 Max. :190.0 Max. :313.00 Max. :215245   
## NA's :259   
## OverallQual OverallCond YearBuilt YearRemodAdd   
## Min. : 1.000 Min. :1.000 Min. :1872 Min. :1950   
## 1st Qu.: 5.000 1st Qu.:5.000 1st Qu.:1954 1st Qu.:1967   
## Median : 6.000 Median :5.000 Median :1973 Median :1994   
## Mean : 6.099 Mean :5.575 Mean :1971 Mean :1985   
## 3rd Qu.: 7.000 3rd Qu.:6.000 3rd Qu.:2000 3rd Qu.:2004   
## Max. :10.000 Max. :9.000 Max. :2010 Max. :2010   
##   
## MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF   
## Min. : 0.0 Min. : 0.0 Min. : 0.00 Min. : 0.0   
## 1st Qu.: 0.0 1st Qu.: 0.0 1st Qu.: 0.00 1st Qu.: 223.0   
## Median : 0.0 Median : 383.5 Median : 0.00 Median : 477.5   
## Mean : 103.7 Mean : 443.6 Mean : 46.55 Mean : 567.2   
## 3rd Qu.: 166.0 3rd Qu.: 712.2 3rd Qu.: 0.00 3rd Qu.: 808.0   
## Max. :1600.0 Max. :5644.0 Max. :1474.00 Max. :2336.0   
## NA's :8   
## TotalBsmtSF 1stFlrSF 2ndFlrSF LowQualFinSF   
## Min. : 0.0 Min. : 334 Min. : 0 Min. : 0.000   
## 1st Qu.: 795.8 1st Qu.: 882 1st Qu.: 0 1st Qu.: 0.000   
## Median : 991.5 Median :1087 Median : 0 Median : 0.000   
## Mean :1057.4 Mean :1163 Mean : 347 Mean : 5.845   
## 3rd Qu.:1298.2 3rd Qu.:1391 3rd Qu.: 728 3rd Qu.: 0.000   
## Max. :6110.0 Max. :4692 Max. :2065 Max. :572.000   
##   
## GrLivArea BsmtFullBath BsmtHalfBath FullBath   
## Min. : 334 Min. :0.0000 Min. :0.00000 Min. :0.000   
## 1st Qu.:1130 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:1.000   
## Median :1464 Median :0.0000 Median :0.00000 Median :2.000   
## Mean :1515 Mean :0.4253 Mean :0.05753 Mean :1.565   
## 3rd Qu.:1777 3rd Qu.:1.0000 3rd Qu.:0.00000 3rd Qu.:2.000   
## Max. :5642 Max. :3.0000 Max. :2.00000 Max. :3.000   
##   
## HalfBath BedroomAbvGr KitchenAbvGr TotRmsAbvGrd   
## Min. :0.0000 Min. :0.000 Min. :0.000 Min. : 2.000   
## 1st Qu.:0.0000 1st Qu.:2.000 1st Qu.:1.000 1st Qu.: 5.000   
## Median :0.0000 Median :3.000 Median :1.000 Median : 6.000   
## Mean :0.3829 Mean :2.866 Mean :1.047 Mean : 6.518   
## 3rd Qu.:1.0000 3rd Qu.:3.000 3rd Qu.:1.000 3rd Qu.: 7.000   
## Max. :2.0000 Max. :8.000 Max. :3.000 Max. :14.000   
##   
## Fireplaces GarageYrBlt GarageCars GarageArea   
## Min. :0.000 Min. :1900 Min. :0.000 Min. : 0.0   
## 1st Qu.:0.000 1st Qu.:1961 1st Qu.:1.000 1st Qu.: 334.5   
## Median :1.000 Median :1980 Median :2.000 Median : 480.0   
## Mean :0.613 Mean :1979 Mean :1.767 Mean : 473.0   
## 3rd Qu.:1.000 3rd Qu.:2002 3rd Qu.:2.000 3rd Qu.: 576.0   
## Max. :3.000 Max. :2010 Max. :4.000 Max. :1418.0   
## NA's :81   
## WoodDeckSF OpenPorchSF EnclosedPorch 3SsnPorch   
## Min. : 0.00 Min. : 0.00 Min. : 0.00 Min. : 0.00   
## 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.: 0.00 1st Qu.: 0.00   
## Median : 0.00 Median : 25.00 Median : 0.00 Median : 0.00   
## Mean : 94.24 Mean : 46.66 Mean : 21.95 Mean : 3.41   
## 3rd Qu.:168.00 3rd Qu.: 68.00 3rd Qu.: 0.00 3rd Qu.: 0.00   
## Max. :857.00 Max. :547.00 Max. :552.00 Max. :508.00   
##   
## ScreenPorch PoolArea MiscVal MoSold   
## Min. : 0.00 Min. : 0.000 Min. : 0.00 Min. : 1.000   
## 1st Qu.: 0.00 1st Qu.: 0.000 1st Qu.: 0.00 1st Qu.: 5.000   
## Median : 0.00 Median : 0.000 Median : 0.00 Median : 6.000   
## Mean : 15.06 Mean : 2.759 Mean : 43.49 Mean : 6.322   
## 3rd Qu.: 0.00 3rd Qu.: 0.000 3rd Qu.: 0.00 3rd Qu.: 8.000   
## Max. :480.00 Max. :738.000 Max. :15500.00 Max. :12.000   
##   
## YrSold SalePrice   
## Min. :2006 Min. : 34900   
## 1st Qu.:2007 1st Qu.:129975   
## Median :2008 Median :163000   
## Mean :2008 Mean :180921   
## 3rd Qu.:2009 3rd Qu.:214000   
## Max. :2010 Max. :755000   
##

nrow(num\_data); ncol(num\_data)

## [1] 1460

## [1] 38

### Low variance variables

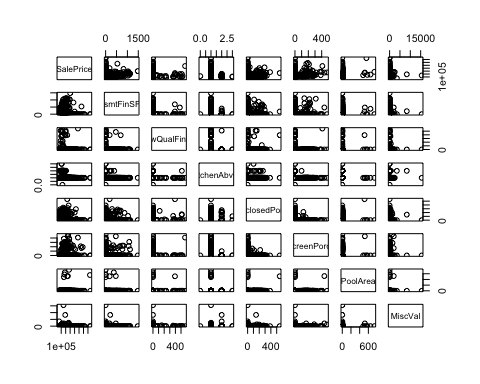
Max Kuhn (2016): Given this, a rule of thumb for detecting near-zero variance predictors is: - The fraction of unique values over the sample size is low (say 10 %). - The ratio of the frequency of the most prevalent value to the frequency of the second most prevalent value is large (say around 20).

#Function calculating the fraction of unique values over the sample size and the ratio of the frequency of the most prevalent value to the frequency of the second most prevalent value  
condition <- function(x) {  
 checking = list() #emty list  
 tbl = sort(table(x), decreasing = TRUE) #Sorting table decreasing  
 checking[["unique\_to\_samp"]] = length(tbl) / sum(tbl) # Get the variance by divide the length of table to sum of the table  
 checking[["most\_prev\_to\_2nd\_prev"]] = (tbl[[1]] / tbl[[2]]) #get ratio  
 checking  
}  
  
#Function checking if unique\_to\_samp < 0.1 and most\_prev\_to\_2nd\_prev >= 20   
low\_var <- function(x) {  
 low\_var\_vec = vector("character", ncol(x))  
 i = 1  
 for (nme in names(x)) {  
 obs = condition(x[[nme]])  
 #print(obs) #test by printing value  
 if (obs[[1]] <= 0.1 & obs[[2]] >= 20) {   
 low\_var\_vec[i] = nme  
 i = i + 1  
 }  
 }  
 low\_var\_vec[low\_var\_vec != ""]  
}  
  
degen\_vec <- low\_var(num\_data); degen\_vec

## [1] "BsmtFinSF2" "LowQualFinSF" "KitchenAbvGr" "EnclosedPorch"  
## [5] "3SsnPorch" "ScreenPorch" "PoolArea" "MiscVal"

num\_data <- select(num\_data, one\_of(setdiff(names(num\_data), degen\_vec))) #make a new dataset without low variance variables

pairs(~SalePrice + BsmtFinSF2+LowQualFinSF + KitchenAbvGr + EnclosedPorch+ ScreenPorch+PoolArea+MiscVal,data=train)



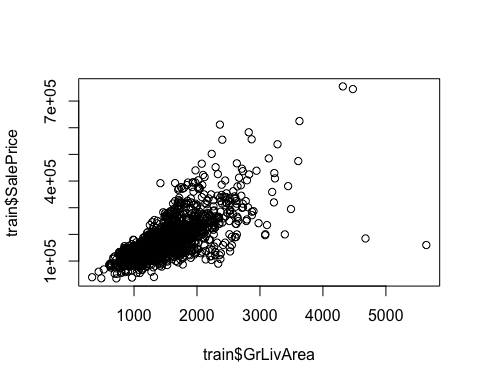
### Multicolinear

The idea is to first remove the predictors that have the most correlated relationships. - Calculate the correlation matrix of the predictors - Determine the two predictors associated with the largest absolute pairwise correlation (call them predictors A and B). - Determine the average correlation between A and the other variables. Do the same for predictor B. - If A has a larger average correlation, remove it; otherwise, remove predictor B. - Repeat Steps 2–4 until no absolute correlations are above the threshold.

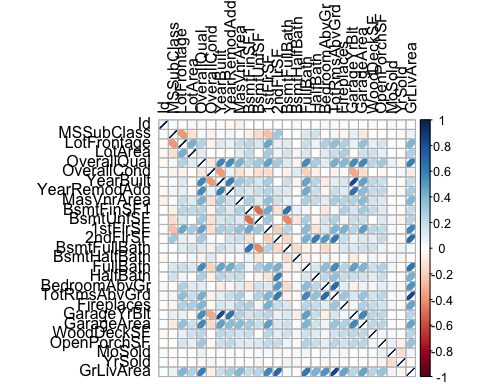
get\_collinear <- function(x) {  
 # Expects data dataframe  
 num\_cols = ncol(x)  
 collinear\_vec = vector("character", num\_cols)   
 index = 1  
   
 for (i in seq(1:num\_cols)) {  
 corMat = cor(x)  
 diag(corMat) = 0 #set diagonal = 0  
 df\_cols = names(x)  
 #Determine the two predictors associated with the largest absolute pairwise correlation (call them predictors A and B).  
 AB = which(corMat == max(abs(corMat), na.rm=TRUE), arr.ind = TRUE)  
 if (corMat[AB][[1]] > 0.75) {  
 names\_AB = rownames(AB)  
   
 if (sum(abs(corMat[names\_AB[1], ]),na.rm=TRUE) > sum(abs(corMat[names\_AB[2], ]),na.rm=TRUE)) {  
   
 collinear\_vec[index] = names\_AB[1]  
 index = index + 1  
 }   
 # if pairwise correlations less than 0.75  
 else {collinear\_vec[index] = names\_AB[2]  
 index = index + 1}  
   
 x = select(x, one\_of(setdiff(df\_cols, collinear\_vec[index - 1])))  
 }  
 else{break}   
 }  
 collinear\_vec[collinear\_vec != ""]  
}  
mul\_col = get\_collinear(num\_data); mul\_col

## [1] "GarageCars" "GrLivArea" "TotalBsmtSF" "SalePrice"

plot(train$GrLivArea, train$SalePrice)



#Correlation matrix for 26 continuous variables  
num\_data <- select(num\_data, one\_of(setdiff(names(num\_data), mul\_col)))  
num\_data$GrLivArea = train$GrLivArea #Adding GrLivArea back  
corrplot(cor(num\_data, use = "pairwise.complete.obs"), method = "ellipse", tl.col = "black", na.label = T)

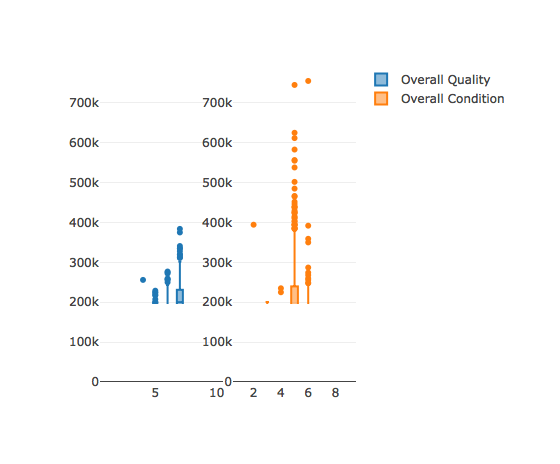


### Decode variables:

copy\_cont\_var = num\_data #copy the continuous dataset to add more variables  
  
#Checking if having lotshape condition is useable or not  
copy\_cont\_var$LotShape\_new <- ifelse(train$LotShape == 'IR3',0,1)  
  
#Checking if having basement exposure or not  
copy\_cont\_var$BsmtExposure\_new <- ifelse(train$BsmtExposure == 'No',0,1)  
copy\_cont\_var$BsmtExposure\_new[is.na(copy\_cont\_var$BsmtExposure\_new)] = 0 #Change NA value = 0

#Checking if having full bath or halfbath  
copy\_cont\_var$FullBath <- ifelse(train$BsmtFullBath > 0,1,0)  
copy\_cont\_var$HalfBath <- ifelse(train$BsmtHalfBath > 0,1,0)  
  
#Checking if having other Miscellaneous or not  
copy\_cont\_var$MiscFeature\_new = ifelse(train$MiscFeature == 'NA',0,1)  
copy\_cont\_var$MiscFeature\_new[is.na(copy\_cont\_var$MiscFeature\_new)] = 0 #Change NA value = 0  
  
#Checking if having fireplace or not  
copy\_cont\_var$Fireplace = ifelse(train$Fireplaces > 0,1,0)  
  
#Checking if having garage or not  
copy\_cont\_var$GarageYrBlt = ifelse(copy\_cont\_var$GarageYrBlt == 'NA',0,1)  
copy\_cont\_var$GarageYrBlt[is.na(copy\_cont\_var$GarageYrBlt)] = 0 #Change NA value = 0  
  
#Checking if having porch/wood desk ...  
copy\_cont\_var$WoodDeckSF <- as.numeric(copy\_cont\_var$WoodDeckSF)  
copy\_cont\_var$OpenPorchSF <- as.numeric(copy\_cont\_var$OpenPorchSF)  
copy\_cont\_var$Porch = copy\_cont\_var$WoodDeckSF + copy\_cont\_var$OpenPorchSF  
copy\_cont\_var$Porch = ifelse(copy\_cont\_var$Porch > 0, 1, 0) #Change to binary var  
  
#Deleting var  
copy\_cont\_var$WoodDeckSF = NULL  
copy\_cont\_var$OpenPorchSF = NULL  
copy\_cont\_var$Id = NULL  
copy\_cont\_var$LotFrontage = NULL  
copy\_cont\_var$YearBuilt = NULL  
copy\_cont\_var$MoSold = NULL

#Relationship between OverallCond vs SalePrice and Overall Quality Condition vs Sale Price  
qual.df <- x\_data[ ,c("OverallQual","OverallCond","SalePrice")]  
pl.q <- plot\_ly(qual.df, y = ~SalePrice, x = ~OverallQual,   
 type = "box", name = "Overall Quality")  
pl.c <- plot\_ly(qual.df, y = ~SalePrice, x = ~OverallCond,   
 type = "box", name = "Overall Condition")  
subplot(pl.q, pl.c)



### Citation

De Cock, Dean. “Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project.” Journal of Statistics Education 19, no. 3 (November 2011). <https://doi.org/10.1080/10691898.2011.11889627>.

Kuhn, Max, and Kjell Johnson. “Data Pre-Processing.” In Applied Predictive Modeling, edited by Max Kuhn and Kjell Johnson, 27–59. New York, NY: Springer New York, 2013. <https://doi.org/10.1007/978-1-4614-6849-3_3>.

“Information About Factors That Determine Property Prices - HomeGuru.” Accessed October 18, 2018. <http://www.homeguru.com.au/house-prices/>.