R Notebook

library(knitr)  
library(ggplot2)  
library(plyr)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(corrplot)

## corrplot 0.84 loaded

library(caret)

## Loading required package: lattice

library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

library(scales)  
library(Rmisc)  
library(ggrepel)  
library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:gridExtra':  
##   
## combine

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(psych)

##   
## Attaching package: 'psych'

## The following object is masked from 'package:randomForest':  
##   
## outlier

## The following objects are masked from 'package:scales':  
##   
## alpha, rescale

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

library(xgboost)

##   
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':  
##   
## slice

test <- read.csv("Data/test.csv", stringsAsFactors = FALSE)  
View(test)  
str(test)

## 'data.frame': 1459 obs. of 80 variables:  
## $ Id : int 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 ...  
## $ MSSubClass : int 20 20 60 60 120 60 20 60 20 20 ...  
## $ MSZoning : chr "RH" "RL" "RL" "RL" ...  
## $ LotFrontage : int 80 81 74 78 43 75 NA 63 85 70 ...  
## $ LotArea : int 11622 14267 13830 9978 5005 10000 7980 8402 10176 8400 ...  
## $ Street : chr "Pave" "Pave" "Pave" "Pave" ...  
## $ Alley : chr NA NA NA NA ...  
## $ LotShape : chr "Reg" "IR1" "IR1" "IR1" ...  
## $ LandContour : chr "Lvl" "Lvl" "Lvl" "Lvl" ...  
## $ Utilities : chr "AllPub" "AllPub" "AllPub" "AllPub" ...  
## $ LotConfig : chr "Inside" "Corner" "Inside" "Inside" ...  
## $ LandSlope : chr "Gtl" "Gtl" "Gtl" "Gtl" ...  
## $ Neighborhood : chr "NAmes" "NAmes" "Gilbert" "Gilbert" ...  
## $ Condition1 : chr "Feedr" "Norm" "Norm" "Norm" ...  
## $ Condition2 : chr "Norm" "Norm" "Norm" "Norm" ...  
## $ BldgType : chr "1Fam" "1Fam" "1Fam" "1Fam" ...  
## $ HouseStyle : chr "1Story" "1Story" "2Story" "2Story" ...  
## $ OverallQual : int 5 6 5 6 8 6 6 6 7 4 ...  
## $ OverallCond : int 6 6 5 6 5 5 7 5 5 5 ...  
## $ YearBuilt : int 1961 1958 1997 1998 1992 1993 1992 1998 1990 1970 ...  
## $ YearRemodAdd : int 1961 1958 1998 1998 1992 1994 2007 1998 1990 1970 ...  
## $ RoofStyle : chr "Gable" "Hip" "Gable" "Gable" ...  
## $ RoofMatl : chr "CompShg" "CompShg" "CompShg" "CompShg" ...  
## $ Exterior1st : chr "VinylSd" "Wd Sdng" "VinylSd" "VinylSd" ...  
## $ Exterior2nd : chr "VinylSd" "Wd Sdng" "VinylSd" "VinylSd" ...  
## $ MasVnrType : chr "None" "BrkFace" "None" "BrkFace" ...  
## $ MasVnrArea : int 0 108 0 20 0 0 0 0 0 0 ...  
## $ ExterQual : chr "TA" "TA" "TA" "TA" ...  
## $ ExterCond : chr "TA" "TA" "TA" "TA" ...  
## $ Foundation : chr "CBlock" "CBlock" "PConc" "PConc" ...  
## $ BsmtQual : chr "TA" "TA" "Gd" "TA" ...  
## $ BsmtCond : chr "TA" "TA" "TA" "TA" ...  
## $ BsmtExposure : chr "No" "No" "No" "No" ...  
## $ BsmtFinType1 : chr "Rec" "ALQ" "GLQ" "GLQ" ...  
## $ BsmtFinSF1 : int 468 923 791 602 263 0 935 0 637 804 ...  
## $ BsmtFinType2 : chr "LwQ" "Unf" "Unf" "Unf" ...  
## $ BsmtFinSF2 : int 144 0 0 0 0 0 0 0 0 78 ...  
## $ BsmtUnfSF : int 270 406 137 324 1017 763 233 789 663 0 ...  
## $ TotalBsmtSF : int 882 1329 928 926 1280 763 1168 789 1300 882 ...  
## $ Heating : chr "GasA" "GasA" "GasA" "GasA" ...  
## $ HeatingQC : chr "TA" "TA" "Gd" "Ex" ...  
## $ CentralAir : chr "Y" "Y" "Y" "Y" ...  
## $ Electrical : chr "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...  
## $ X1stFlrSF : int 896 1329 928 926 1280 763 1187 789 1341 882 ...  
## $ X2ndFlrSF : int 0 0 701 678 0 892 0 676 0 0 ...  
## $ LowQualFinSF : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ GrLivArea : int 896 1329 1629 1604 1280 1655 1187 1465 1341 882 ...  
## $ BsmtFullBath : int 0 0 0 0 0 0 1 0 1 1 ...  
## $ BsmtHalfBath : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ FullBath : int 1 1 2 2 2 2 2 2 1 1 ...  
## $ HalfBath : int 0 1 1 1 0 1 0 1 1 0 ...  
## $ BedroomAbvGr : int 2 3 3 3 2 3 3 3 2 2 ...  
## $ KitchenAbvGr : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ KitchenQual : chr "TA" "Gd" "TA" "Gd" ...  
## $ TotRmsAbvGrd : int 5 6 6 7 5 7 6 7 5 4 ...  
## $ Functional : chr "Typ" "Typ" "Typ" "Typ" ...  
## $ Fireplaces : int 0 0 1 1 0 1 0 1 1 0 ...  
## $ FireplaceQu : chr NA NA "TA" "Gd" ...  
## $ GarageType : chr "Attchd" "Attchd" "Attchd" "Attchd" ...  
## $ GarageYrBlt : int 1961 1958 1997 1998 1992 1993 1992 1998 1990 1970 ...  
## $ GarageFinish : chr "Unf" "Unf" "Fin" "Fin" ...  
## $ GarageCars : int 1 1 2 2 2 2 2 2 2 2 ...  
## $ GarageArea : int 730 312 482 470 506 440 420 393 506 525 ...  
## $ GarageQual : chr "TA" "TA" "TA" "TA" ...  
## $ GarageCond : chr "TA" "TA" "TA" "TA" ...  
## $ PavedDrive : chr "Y" "Y" "Y" "Y" ...  
## $ WoodDeckSF : int 140 393 212 360 0 157 483 0 192 240 ...  
## $ OpenPorchSF : int 0 36 34 36 82 84 21 75 0 0 ...  
## $ EnclosedPorch: int 0 0 0 0 0 0 0 0 0 0 ...  
## $ X3SsnPorch : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ ScreenPorch : int 120 0 0 0 144 0 0 0 0 0 ...  
## $ PoolArea : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ PoolQC : chr NA NA NA NA ...  
## $ Fence : chr "MnPrv" NA "MnPrv" NA ...  
## $ MiscFeature : chr NA "Gar2" NA NA ...  
## $ MiscVal : int 0 12500 0 0 0 0 500 0 0 0 ...  
## $ MoSold : int 6 6 3 6 1 4 3 5 2 4 ...  
## $ YrSold : int 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...  
## $ SaleType : chr "WD" "WD" "WD" "WD" ...  
## $ SaleCondition: chr "Normal" "Normal" "Normal" "Normal" ...

train <- read.csv("Data/train.csv", stringsAsFactors = FALSE)  
View(train)  
str(train)

## 'data.frame': 1460 obs. of 81 variables:  
## $ Id : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ MSSubClass : int 60 20 60 70 60 50 20 60 50 190 ...  
## $ MSZoning : chr "RL" "RL" "RL" "RL" ...  
## $ LotFrontage : int 65 80 68 60 84 85 75 NA 51 50 ...  
## $ LotArea : int 8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...  
## $ Street : chr "Pave" "Pave" "Pave" "Pave" ...  
## $ Alley : chr NA NA NA NA ...  
## $ LotShape : chr "Reg" "Reg" "IR1" "IR1" ...  
## $ LandContour : chr "Lvl" "Lvl" "Lvl" "Lvl" ...  
## $ Utilities : chr "AllPub" "AllPub" "AllPub" "AllPub" ...  
## $ LotConfig : chr "Inside" "FR2" "Inside" "Corner" ...  
## $ LandSlope : chr "Gtl" "Gtl" "Gtl" "Gtl" ...  
## $ Neighborhood : chr "CollgCr" "Veenker" "CollgCr" "Crawfor" ...  
## $ Condition1 : chr "Norm" "Feedr" "Norm" "Norm" ...  
## $ Condition2 : chr "Norm" "Norm" "Norm" "Norm" ...  
## $ BldgType : chr "1Fam" "1Fam" "1Fam" "1Fam" ...  
## $ HouseStyle : chr "2Story" "1Story" "2Story" "2Story" ...  
## $ OverallQual : int 7 6 7 7 8 5 8 7 7 5 ...  
## $ OverallCond : int 5 8 5 5 5 5 5 6 5 6 ...  
## $ YearBuilt : int 2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...  
## $ YearRemodAdd : int 2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...  
## $ RoofStyle : chr "Gable" "Gable" "Gable" "Gable" ...  
## $ RoofMatl : chr "CompShg" "CompShg" "CompShg" "CompShg" ...  
## $ Exterior1st : chr "VinylSd" "MetalSd" "VinylSd" "Wd Sdng" ...  
## $ Exterior2nd : chr "VinylSd" "MetalSd" "VinylSd" "Wd Shng" ...  
## $ MasVnrType : chr "BrkFace" "None" "BrkFace" "None" ...  
## $ MasVnrArea : int 196 0 162 0 350 0 186 240 0 0 ...  
## $ ExterQual : chr "Gd" "TA" "Gd" "TA" ...  
## $ ExterCond : chr "TA" "TA" "TA" "TA" ...  
## $ Foundation : chr "PConc" "CBlock" "PConc" "BrkTil" ...  
## $ BsmtQual : chr "Gd" "Gd" "Gd" "TA" ...  
## $ BsmtCond : chr "TA" "TA" "TA" "Gd" ...  
## $ BsmtExposure : chr "No" "Gd" "Mn" "No" ...  
## $ BsmtFinType1 : chr "GLQ" "ALQ" "GLQ" "ALQ" ...  
## $ BsmtFinSF1 : int 706 978 486 216 655 732 1369 859 0 851 ...  
## $ BsmtFinType2 : chr "Unf" "Unf" "Unf" "Unf" ...  
## $ BsmtFinSF2 : int 0 0 0 0 0 0 0 32 0 0 ...  
## $ BsmtUnfSF : int 150 284 434 540 490 64 317 216 952 140 ...  
## $ TotalBsmtSF : int 856 1262 920 756 1145 796 1686 1107 952 991 ...  
## $ Heating : chr "GasA" "GasA" "GasA" "GasA" ...  
## $ HeatingQC : chr "Ex" "Ex" "Ex" "Gd" ...  
## $ CentralAir : chr "Y" "Y" "Y" "Y" ...  
## $ Electrical : chr "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...  
## $ X1stFlrSF : int 856 1262 920 961 1145 796 1694 1107 1022 1077 ...  
## $ X2ndFlrSF : int 854 0 866 756 1053 566 0 983 752 0 ...  
## $ LowQualFinSF : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ GrLivArea : int 1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...  
## $ BsmtFullBath : int 1 0 1 1 1 1 1 1 0 1 ...  
## $ BsmtHalfBath : int 0 1 0 0 0 0 0 0 0 0 ...  
## $ FullBath : int 2 2 2 1 2 1 2 2 2 1 ...  
## $ HalfBath : int 1 0 1 0 1 1 0 1 0 0 ...  
## $ BedroomAbvGr : int 3 3 3 3 4 1 3 3 2 2 ...  
## $ KitchenAbvGr : int 1 1 1 1 1 1 1 1 2 2 ...  
## $ KitchenQual : chr "Gd" "TA" "Gd" "Gd" ...  
## $ TotRmsAbvGrd : int 8 6 6 7 9 5 7 7 8 5 ...  
## $ Functional : chr "Typ" "Typ" "Typ" "Typ" ...  
## $ Fireplaces : int 0 1 1 1 1 0 1 2 2 2 ...  
## $ FireplaceQu : chr NA "TA" "TA" "Gd" ...  
## $ GarageType : chr "Attchd" "Attchd" "Attchd" "Detchd" ...  
## $ GarageYrBlt : int 2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 ...  
## $ GarageFinish : chr "RFn" "RFn" "RFn" "Unf" ...  
## $ GarageCars : int 2 2 2 3 3 2 2 2 2 1 ...  
## $ GarageArea : int 548 460 608 642 836 480 636 484 468 205 ...  
## $ GarageQual : chr "TA" "TA" "TA" "TA" ...  
## $ GarageCond : chr "TA" "TA" "TA" "TA" ...  
## $ PavedDrive : chr "Y" "Y" "Y" "Y" ...  
## $ WoodDeckSF : int 0 298 0 0 192 40 255 235 90 0 ...  
## $ OpenPorchSF : int 61 0 42 35 84 30 57 204 0 4 ...  
## $ EnclosedPorch: int 0 0 0 272 0 0 0 228 205 0 ...  
## $ X3SsnPorch : int 0 0 0 0 0 320 0 0 0 0 ...  
## $ ScreenPorch : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ PoolArea : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ PoolQC : chr NA NA NA NA ...  
## $ Fence : chr NA NA NA NA ...  
## $ MiscFeature : chr NA NA NA NA ...  
## $ MiscVal : int 0 0 0 0 0 700 0 350 0 0 ...  
## $ MoSold : int 2 5 9 2 12 10 8 11 4 1 ...  
## $ YrSold : int 2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...  
## $ SaleType : chr "WD" "WD" "WD" "WD" ...  
## $ SaleCondition: chr "Normal" "Normal" "Normal" "Abnorml" ...  
## $ SalePrice : int 208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...

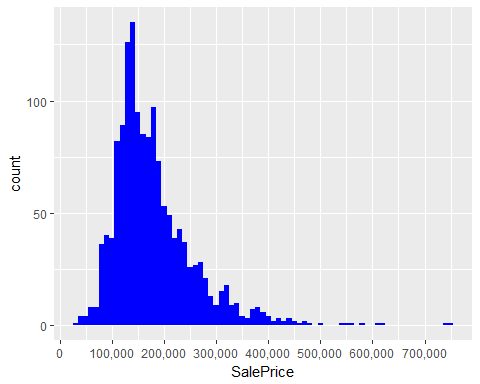
##Combine Train & Test into singular dataframe by dropping ID column  
test.IDs <- test$Id  
test$Id <- NULL  
train$Id <- NULL  
  
test$SalePrice <- NA  
df.combined <- rbind(train, test)  
str(df.combined)

## 'data.frame': 2919 obs. of 80 variables:  
## $ MSSubClass : int 60 20 60 70 60 50 20 60 50 190 ...  
## $ MSZoning : chr "RL" "RL" "RL" "RL" ...  
## $ LotFrontage : int 65 80 68 60 84 85 75 NA 51 50 ...  
## $ LotArea : int 8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...  
## $ Street : chr "Pave" "Pave" "Pave" "Pave" ...  
## $ Alley : chr NA NA NA NA ...  
## $ LotShape : chr "Reg" "Reg" "IR1" "IR1" ...  
## $ LandContour : chr "Lvl" "Lvl" "Lvl" "Lvl" ...  
## $ Utilities : chr "AllPub" "AllPub" "AllPub" "AllPub" ...  
## $ LotConfig : chr "Inside" "FR2" "Inside" "Corner" ...  
## $ LandSlope : chr "Gtl" "Gtl" "Gtl" "Gtl" ...  
## $ Neighborhood : chr "CollgCr" "Veenker" "CollgCr" "Crawfor" ...  
## $ Condition1 : chr "Norm" "Feedr" "Norm" "Norm" ...  
## $ Condition2 : chr "Norm" "Norm" "Norm" "Norm" ...  
## $ BldgType : chr "1Fam" "1Fam" "1Fam" "1Fam" ...  
## $ HouseStyle : chr "2Story" "1Story" "2Story" "2Story" ...  
## $ OverallQual : int 7 6 7 7 8 5 8 7 7 5 ...  
## $ OverallCond : int 5 8 5 5 5 5 5 6 5 6 ...  
## $ YearBuilt : int 2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...  
## $ YearRemodAdd : int 2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...  
## $ RoofStyle : chr "Gable" "Gable" "Gable" "Gable" ...  
## $ RoofMatl : chr "CompShg" "CompShg" "CompShg" "CompShg" ...  
## $ Exterior1st : chr "VinylSd" "MetalSd" "VinylSd" "Wd Sdng" ...  
## $ Exterior2nd : chr "VinylSd" "MetalSd" "VinylSd" "Wd Shng" ...  
## $ MasVnrType : chr "BrkFace" "None" "BrkFace" "None" ...  
## $ MasVnrArea : int 196 0 162 0 350 0 186 240 0 0 ...  
## $ ExterQual : chr "Gd" "TA" "Gd" "TA" ...  
## $ ExterCond : chr "TA" "TA" "TA" "TA" ...  
## $ Foundation : chr "PConc" "CBlock" "PConc" "BrkTil" ...  
## $ BsmtQual : chr "Gd" "Gd" "Gd" "TA" ...  
## $ BsmtCond : chr "TA" "TA" "TA" "Gd" ...  
## $ BsmtExposure : chr "No" "Gd" "Mn" "No" ...  
## $ BsmtFinType1 : chr "GLQ" "ALQ" "GLQ" "ALQ" ...  
## $ BsmtFinSF1 : int 706 978 486 216 655 732 1369 859 0 851 ...  
## $ BsmtFinType2 : chr "Unf" "Unf" "Unf" "Unf" ...  
## $ BsmtFinSF2 : int 0 0 0 0 0 0 0 32 0 0 ...  
## $ BsmtUnfSF : int 150 284 434 540 490 64 317 216 952 140 ...  
## $ TotalBsmtSF : int 856 1262 920 756 1145 796 1686 1107 952 991 ...  
## $ Heating : chr "GasA" "GasA" "GasA" "GasA" ...  
## $ HeatingQC : chr "Ex" "Ex" "Ex" "Gd" ...  
## $ CentralAir : chr "Y" "Y" "Y" "Y" ...  
## $ Electrical : chr "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...  
## $ X1stFlrSF : int 856 1262 920 961 1145 796 1694 1107 1022 1077 ...  
## $ X2ndFlrSF : int 854 0 866 756 1053 566 0 983 752 0 ...  
## $ LowQualFinSF : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ GrLivArea : int 1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...  
## $ BsmtFullBath : int 1 0 1 1 1 1 1 1 0 1 ...  
## $ BsmtHalfBath : int 0 1 0 0 0 0 0 0 0 0 ...  
## $ FullBath : int 2 2 2 1 2 1 2 2 2 1 ...  
## $ HalfBath : int 1 0 1 0 1 1 0 1 0 0 ...  
## $ BedroomAbvGr : int 3 3 3 3 4 1 3 3 2 2 ...  
## $ KitchenAbvGr : int 1 1 1 1 1 1 1 1 2 2 ...  
## $ KitchenQual : chr "Gd" "TA" "Gd" "Gd" ...  
## $ TotRmsAbvGrd : int 8 6 6 7 9 5 7 7 8 5 ...  
## $ Functional : chr "Typ" "Typ" "Typ" "Typ" ...  
## $ Fireplaces : int 0 1 1 1 1 0 1 2 2 2 ...  
## $ FireplaceQu : chr NA "TA" "TA" "Gd" ...  
## $ GarageType : chr "Attchd" "Attchd" "Attchd" "Detchd" ...  
## $ GarageYrBlt : int 2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 ...  
## $ GarageFinish : chr "RFn" "RFn" "RFn" "Unf" ...  
## $ GarageCars : int 2 2 2 3 3 2 2 2 2 1 ...  
## $ GarageArea : int 548 460 608 642 836 480 636 484 468 205 ...  
## $ GarageQual : chr "TA" "TA" "TA" "TA" ...  
## $ GarageCond : chr "TA" "TA" "TA" "TA" ...  
## $ PavedDrive : chr "Y" "Y" "Y" "Y" ...  
## $ WoodDeckSF : int 0 298 0 0 192 40 255 235 90 0 ...  
## $ OpenPorchSF : int 61 0 42 35 84 30 57 204 0 4 ...  
## $ EnclosedPorch: int 0 0 0 272 0 0 0 228 205 0 ...  
## $ X3SsnPorch : int 0 0 0 0 0 320 0 0 0 0 ...  
## $ ScreenPorch : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ PoolArea : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ PoolQC : chr NA NA NA NA ...  
## $ Fence : chr NA NA NA NA ...  
## $ MiscFeature : chr NA NA NA NA ...  
## $ MiscVal : int 0 0 0 0 0 700 0 350 0 0 ...  
## $ MoSold : int 2 5 9 2 12 10 8 11 4 1 ...  
## $ YrSold : int 2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...  
## $ SaleType : chr "WD" "WD" "WD" "WD" ...  
## $ SaleCondition: chr "Normal" "Normal" "Normal" "Abnorml" ...  
## $ SalePrice : int 208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...

##Visualize distribution of SalePrice Values  
summary(df.combined$SalePrice)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 34900 129975 163000 180921 214000 755000 1459

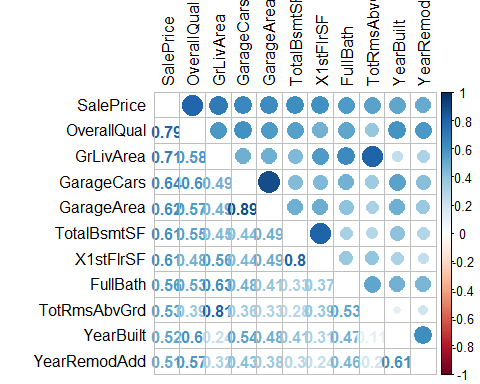
ggplot(data = df.combined[!is.na(df.combined$SalePrice),], aes(x = SalePrice)) +  
 geom\_histogram(fill = "blue", binwidth = 10000) +  
 scale\_x\_continuous(breaks = seq(0, 800000, by = 100000), labels = scales::comma)



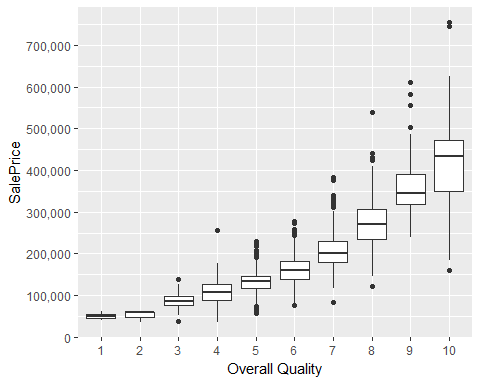
##Index Vector of Numeric Variables  
numeric.Vars <- which(sapply(df.combined, is.numeric))   
##Names Vector of Numeric Variables  
numeric.VarNames <- names(numeric.Vars)  
  
cat(length(numeric.Vars), 'numeric variables')

## 37 numeric variables

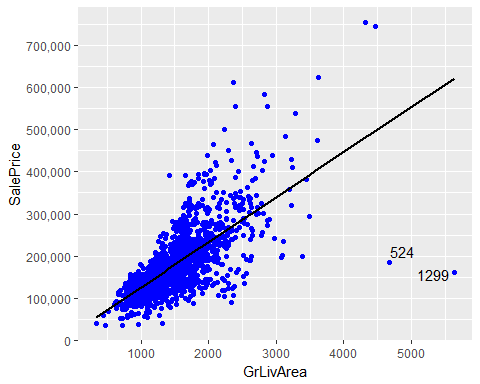
##Dataframe of Numeric Variables  
df.numeric.Vars <- df.combined[, numeric.Vars]  
##Correlation of Numeric Variables  
correlation.numeric.Vars <- cor(df.numeric.Vars, use = "pairwise.complete.obs")  
##Sort on decreasing correlations by SalePrice  
correlation.sorted <- as.matrix(sort(correlation.numeric.Vars[, 'SalePrice'], decreasing = TRUE))  
##Select High Correlations  
correlation.high <- names(which(apply(correlation.sorted, 1, function(x) abs(x) > 0.5)))  
correlation.numeric.Vars <- correlation.numeric.Vars[correlation.high, correlation.high]  
##Correlation Plot  
corrplot.mixed(correlation.numeric.Vars, tl.col="black", tl.pos = "lt")



##SalePrice vs. Overall Quality (Highest Correlation to SalePrice)  
ggplot(data = df.combined[!is.na(df.combined$SalePrice),], aes(x = factor(OverallQual), y = SalePrice))+  
 geom\_boxplot() + labs(x = 'Overall Quality') +  
 scale\_y\_continuous(breaks = seq(0, 800000, by = 100000), labels = scales::comma)



##SalePrice vs. GrLivArea (2nd Highest Correlation to SalePrice)  
ggplot(data = df.combined[!is.na(df.combined$SalePrice),], aes(x = GrLivArea, y = SalePrice)) +  
 geom\_point(col = 'blue') + geom\_smooth(method = "lm", se = FALSE, color = "black", aes(group = 1)) +  
 scale\_y\_continuous(breaks = seq(0, 800000, by = 100000), labels = scales::comma) +  
 geom\_text\_repel(aes(label = ifelse(df.combined$GrLivArea[!is.na(df.combined$SalePrice)] > 4500, rownames(df.combined), '')))



##Low SalePrice, High Quality Outliers  
df.combined[c(524, 1299), c('SalePrice', 'GrLivArea', 'OverallQual')]

## SalePrice GrLivArea OverallQual  
## 524 184750 4676 10  
## 1299 160000 5642 10

##Variables with Missing Values  
NAcol <- which(colSums(is.na(df.combined)) > 0)  
sort(colSums(sapply(df.combined[NAcol], is.na)), decreasing = TRUE)

## PoolQC MiscFeature Alley Fence SalePrice   
## 2909 2814 2721 2348 1459   
## FireplaceQu LotFrontage GarageYrBlt GarageFinish GarageQual   
## 1420 486 159 159 159   
## GarageCond GarageType BsmtCond BsmtExposure BsmtQual   
## 159 157 82 82 81   
## BsmtFinType2 BsmtFinType1 MasVnrType MasVnrArea MSZoning   
## 80 79 24 23 4   
## Utilities BsmtFullBath BsmtHalfBath Functional Exterior1st   
## 2 2 2 2 1   
## Exterior2nd BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF   
## 1 1 1 1 1   
## Electrical KitchenQual GarageCars GarageArea SaleType   
## 1 1 1 1 1

cat(length(NAcol), 'variables with missing values')

## 35 variables with missing values

##Imputing Missing Pool Values  
unique(df.combined$PoolQC)

## [1] NA "Ex" "Fa" "Gd"

df.combined$PoolQC[is.na(df.combined$PoolQC)] <- 'None'  
  
##Quality Level Vector  
Qualities <- c('None' = 0, 'Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)  
  
##Impute PoolQC with Quality Level Vector  
df.combined$PoolQC <- as.integer(revalue(df.combined$PoolQC, Qualities))

## The following `from` values were not present in `x`: Po, TA

table(df.combined$PoolQC)

##   
## 0 2 4 5   
## 2909 2 4 4

##Verify 3 values without PoolQC  
df.combined[df.combined$PoolArea > 0 & df.combined$PoolQC == 0, c('PoolArea', 'PoolQC', 'OverallQual')]

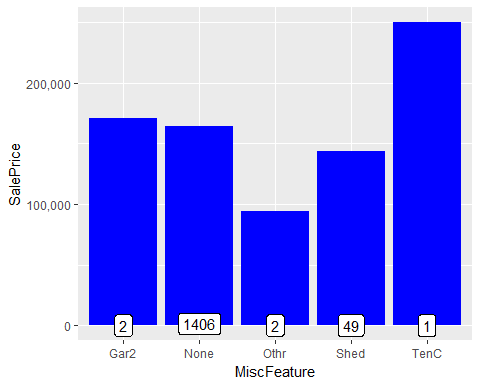
## PoolArea PoolQC OverallQual  
## 2421 368 0 4  
## 2504 444 0 6  
## 2600 561 0 3

##Impute 3 values without PoolQC with OverallQual  
df.combined$PoolQC[2421] <- 2  
df.combined$PoolQC[2504] <- 3  
df.combined$PoolQC[2600] <- 2

##Imputing Missing MiscFeature Values  
unique(df.combined$MiscFeature)

## [1] NA "Shed" "Gar2" "Othr" "TenC"

df.combined$MiscFeature[is.na(df.combined$MiscFeature)] <- 'None'  
df.combined$MiscFeature <- as.factor(df.combined$MiscFeature)  
  
ggplot(df.combined[!is.na(df.combined$SalePrice),], aes(x = MiscFeature, y = SalePrice)) +  
 geom\_bar(stat = 'summary', fun.y = "median", fill = 'blue') +  
 scale\_y\_continuous(breaks = seq(0, 800000, by = 100000), labels = scales::comma) +  
 geom\_label(stat = "count", aes(label = ..count.., y = ..count..))



table(df.combined$MiscFeature)

##   
## Gar2 None Othr Shed TenC   
## 5 2814 4 95 1

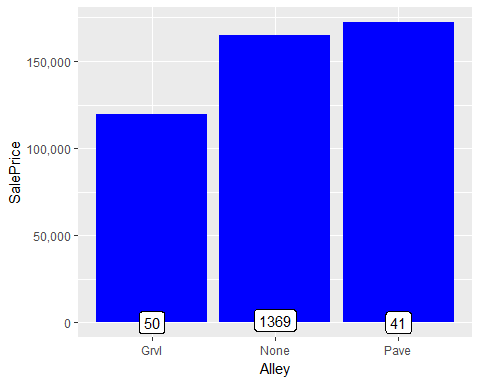
##Imputing Missing Alley Values  
unique(df.combined$Alley)

## [1] NA "Grvl" "Pave"

df.combined$Alley[is.na(df.combined$Alley)] <- 'None'  
df.combined$Alley <- as.factor(df.combined$Alley)  
table(df.combined$Alley)

##   
## Grvl None Pave   
## 120 2721 78

ggplot(df.combined[!is.na(df.combined$SalePrice),], aes(x = Alley, y = SalePrice)) +  
 geom\_bar(stat = 'summary', fun.y = "median", fill = 'blue')+  
 scale\_y\_continuous(breaks = seq(0, 200000, by = 50000), labels = scales::comma) +  
 geom\_label(stat = "count", aes(label = ..count.., y = ..count..))



##Imputing Missing Fence Values  
unique(df.combined$Fence)

## [1] NA "MnPrv" "GdWo" "GdPrv" "MnWw"

df.combined$Fence[is.na(df.combined$Fence)] <- 'None'  
table(df.combined$Fence)

##   
## GdPrv GdWo MnPrv MnWw None   
## 118 112 329 12 2348

df.combined[!is.na(df.combined$SalePrice),] %>%   
 group\_by(Fence) %>%   
 summarise(median = median(SalePrice), counts=n())

## # A tibble: 5 x 3  
## Fence median counts  
## <chr> <dbl> <int>  
## 1 GdPrv 167500 59  
## 2 GdWo 138750 54  
## 3 MnPrv 137450 157  
## 4 MnWw 130000 11  
## 5 None 173000 1179

df.combined$Fence <- as.factor(df.combined$Fence)

##Imputing Missing Fireplace Values  
  
##FireplaceQu  
unique(df.combined$FireplaceQu)

## [1] NA "TA" "Gd" "Fa" "Ex" "Po"

df.combined$FireplaceQu[is.na(df.combined$FireplaceQu)] <- 'None'  
df.combined$FireplaceQu <- as.integer(revalue(df.combined$FireplaceQu, Qualities))  
table(df.combined$FireplaceQu)

##   
## 0 1 2 3 4 5   
## 1420 46 74 592 744 43

##Fireplaces  
unique(df.combined$Fireplaces)

## [1] 0 1 2 3 4

table(df.combined$Fireplaces)

##   
## 0 1 2 3 4   
## 1420 1268 219 11 1

sum(table(df.combined$Fireplaces))

## [1] 2919

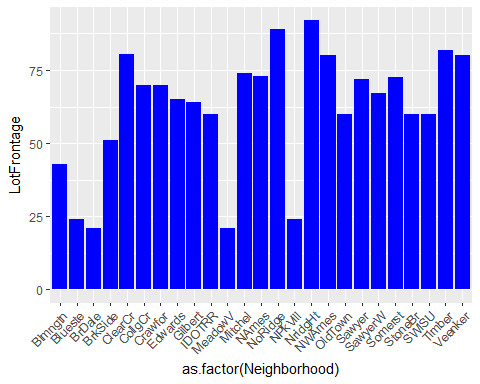
##Imputing Missing Lot Values  
  
##LotFrontage  
unique(df.combined$LotFrontage)

## [1] 65 80 68 60 84 85 75 NA 51 50 70 91 72 66 101 57 44  
## [18] 110 98 47 108 112 74 115 61 48 33 52 100 24 89 63 76 81  
## [35] 95 69 21 32 78 121 122 40 105 73 77 64 94 34 90 55 88  
## [52] 82 71 120 107 92 134 62 86 141 97 54 41 79 174 99 67 83  
## [69] 43 103 93 30 129 140 35 37 118 87 116 150 111 49 96 59 36  
## [86] 56 102 58 38 109 130 53 137 45 106 104 42 39 144 114 128 149  
## [103] 313 168 182 138 160 152 124 153 46 26 25 119 31 28 117 113 125  
## [120] 135 136 22 123 195 155 126 200 131 133

table(is.na(df.combined$LotFrontage))

##   
## FALSE TRUE   
## 2433 486

ggplot(df.combined[!is.na(df.combined$LotFrontage),], aes(x = as.factor(Neighborhood), y = LotFrontage)) +  
 geom\_bar(stat = 'summary', fun.y = "median", fill = 'blue') +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



for (i in 1:nrow(df.combined)){  
 if(is.na(df.combined$LotFrontage[i])){  
 df.combined$LotFrontage[i] <- as.integer(median(df.combined$LotFrontage[df.combined$Neighborhood == df.combined$Neighborhood[i]], na.rm = TRUE))   
 }  
}  
  
##LotShape  
unique(df.combined$LotShape)

## [1] "Reg" "IR1" "IR2" "IR3"

df.combined$LotShape <- as.integer(revalue(df.combined$LotShape, c('IR3' = 0, 'IR2' = 1, 'IR1' = 2, 'Reg'= 3)))  
table(df.combined$LotShape)

##   
## 0 1 2 3   
## 16 76 968 1859

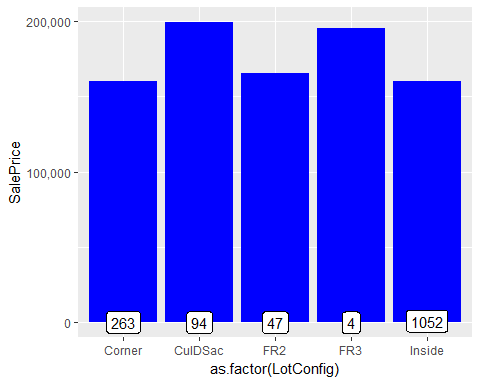
sum(table(df.combined$LotShape))

## [1] 2919

##LotConfig  
unique(df.combined$LotConfig)

## [1] "Inside" "FR2" "Corner" "CulDSac" "FR3"

ggplot(df.combined[!is.na(df.combined$SalePrice),], aes(x = as.factor(LotConfig), y = SalePrice)) +  
 geom\_bar(stat = 'summary', fun.y = "median", fill = 'blue')+  
 scale\_y\_continuous(breaks = seq(0, 800000, by = 100000), labels = scales::comma) +  
 geom\_label(stat = "count", aes(label = ..count.., y = ..count..))



df.combined$LotConfig <- as.factor(df.combined$LotConfig)  
table(df.combined$LotConfig)

##   
## Corner CulDSac FR2 FR3 Inside   
## 511 176 85 14 2133

sum(table(df.combined$LotConfig))

## [1] 2919

##Imputing Missing Garage Values  
unique(df.combined$GarageYrBlt)

## [1] 2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 1965 2005 1962 2006  
## [15] 1960 1991 1970 1967 1958 1930 2002 1968 2007 2008 1957 1920 1966 1959  
## [29] 1995 1954 1953 NA 1983 1977 1997 1985 1963 1981 1964 1999 1935 1990  
## [43] 1945 1987 1989 1915 1956 1948 1974 2009 1950 1961 1921 1900 1979 1951  
## [57] 1969 1936 1975 1971 1923 1984 1926 1955 1986 1988 1916 1932 1972 1918  
## [71] 1980 1924 1996 1940 1949 1994 1910 1978 1982 1992 1925 1941 2010 1927  
## [85] 1947 1937 1942 1938 1952 1928 1922 1934 1906 1914 1946 1908 1929 1933  
## [99] 1917 1896 1895 2207 1943 1919

table(is.na(df.combined$GarageYrBlt))

##   
## FALSE TRUE   
## 2760 159

##159 NA's - Replacing GarageYrBlt with YearBuilt  
df.combined$GarageYrBlt[is.na(df.combined$GarageYrBlt)] <- df.combined$YearBuilt[is.na(df.combined$GarageYrBlt)]  
  
##157 NA's imputed - 2 have Garage Values but are NA  
length(which(is.na(df.combined$GarageType) & is.na(df.combined$GarageFinish) & is.na(df.combined$GarageCond) & is.na(df.combined$GarageQual)))

## [1] 157

##Identify and validate 2 NA's with Garage Values  
kable(df.combined[!is.na(df.combined$GarageType) & is.na(df.combined$GarageFinish), c('GarageCars', 'GarageArea', 'GarageType', 'GarageCond', 'GarageQual', 'GarageFinish')])

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | GarageCars | GarageArea | GarageType | GarageCond | GarageQual | GarageFinish |
| 2127 | 1 | 360 | Detchd | NA | NA | NA |
| 2577 | NA | NA | Detchd | NA | NA | NA |

##Imputing House #2127 Missing Garage Values with modes  
df.combined$GarageCond[2127] <- names(sort(-table(df.combined$GarageCond)))[1]  
df.combined$GarageQual[2127] <- names(sort(-table(df.combined$GarageQual)))[1]  
df.combined$GarageFinish[2127] <- names(sort(-table(df.combined$GarageFinish)))[1]  
  
##Check House #2127  
kable(df.combined[2127, c('GarageYrBlt', 'GarageCars', 'GarageArea', 'GarageType', 'GarageCond', 'GarageQual', 'GarageFinish')])

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | GarageYrBlt | GarageCars | GarageArea | GarageType | GarageCond | GarageQual | GarageFinish |
| 2127 | 1910 | 1 | 360 | Detchd | TA | TA | Unf |

##Fix values for House ##2577  
df.combined$GarageCars[2577] <- 0  
df.combined$GarageArea[2577] <- 0  
df.combined$GarageType[2577] <- NA  
  
##Check House #2577  
kable(df.combined[2577, c('GarageYrBlt', 'GarageCars', 'GarageArea', 'GarageType', 'GarageCond', 'GarageQual', 'GarageFinish')])

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | GarageYrBlt | GarageCars | GarageArea | GarageType | GarageCond | GarageQual | GarageFinish |
| 2577 | 1923 | 0 | 0 | NA | NA | NA | NA |

##Verify that there are 158 homes without garages  
length(which(is.na(df.combined$GarageType) & is.na(df.combined$GarageFinish) & is.na(df.combined$GarageCond) & is.na(df.combined$GarageQual)))

## [1] 158

##Imputing Missing GarageType Values  
unique(df.combined$GarageType)

## [1] "Attchd" "Detchd" "BuiltIn" "CarPort" NA "Basment" "2Types"

df.combined$GarageType[is.na(df.combined$GarageType)] <- 'No Garage'  
df.combined$GarageType <- as.factor(df.combined$GarageType)  
table(df.combined$GarageType)

##   
## 2Types Attchd Basment BuiltIn CarPort Detchd No Garage   
## 23 1723 36 186 15 778 158

##Imputing Missing GarageFinish Values  
unique(df.combined$GarageFinish)

## [1] "RFn" "Unf" "Fin" NA

df.combined$GarageFinish[is.na(df.combined$GarageFinish)] <- 'None'  
  
##Finish Level Vector  
Finish <- c('None' = 0, 'Unf' = 1, 'RFn' = 2, 'Fin' = 3)  
  
df.combined$GarageFinish <- as.integer(revalue(df.combined$GarageFinish, Finish))  
table(df.combined$GarageFinish)

##   
## 0 1 2 3   
## 158 1231 811 719

##Imputing Missing GarageQual Values  
unique(df.combined$GarageQual)

## [1] "TA" "Fa" "Gd" NA "Ex" "Po"

df.combined$GarageQual[is.na(df.combined$GarageQual)] <- 'None'  
df.combined$GarageQual <- as.integer(revalue(df.combined$GarageQual, Qualities))  
table(df.combined$GarageQual)

##   
## 0 1 2 3 4 5   
## 158 5 124 2605 24 3

##Imputing Missing GarageCond Values  
unique(df.combined$GarageCond)

## [1] "TA" "Fa" NA "Gd" "Po" "Ex"

df.combined$GarageCond[is.na(df.combined$GarageCond)] <- 'None'  
df.combined$GarageCond<-as.integer(revalue(df.combined$GarageCond, Qualities))  
table(df.combined$GarageCond)

##   
## 0 1 2 3 4 5   
## 158 14 74 2655 15 3

##Imputing Missing Basement Values  
unique(df.combined$BsmtQual)

## [1] "Gd" "TA" "Ex" NA "Fa"

table(is.na(df.combined$BsmtQual))

##   
## FALSE TRUE   
## 2838 81

##Verifying set of 79 NA obserations  
length(which(is.na(df.combined$BsmtQual) & is.na(df.combined$BsmtCond) & is.na(df.combined$BsmtExposure) & is.na(df.combined$BsmtFinType1) & is.na(df.combined$BsmtFinType2)))

## [1] 79

##Find overlaps of NA's between Basement Variables   
df.combined[!is.na(df.combined$BsmtFinType1) & (is.na(df.combined$BsmtCond)|is.na(df.combined$BsmtQual)|is.na(df.combined$BsmtExposure)|is.na(df.combined$BsmtFinType2)), c('BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2')]

## BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinType2  
## 333 Gd TA No GLQ <NA>  
## 949 Gd TA <NA> Unf Unf  
## 1488 Gd TA <NA> Unf Unf  
## 2041 Gd <NA> Mn GLQ Rec  
## 2186 TA <NA> No BLQ Unf  
## 2218 <NA> Fa No Unf Unf  
## 2219 <NA> TA No Unf Unf  
## 2349 Gd TA <NA> Unf Unf  
## 2525 TA <NA> Av ALQ Unf

##Imputing Modes of Bsmt Values  
df.combined$BsmtFinType2[333] <- names(sort(-table(df.combined$BsmtFinType2)))[1]  
df.combined$BsmtExposure[c(949, 1488, 2349)] <- names(sort(-table(df.combined$BsmtExposure)))[1]  
df.combined$BsmtCond[c(2041, 2186, 2525)] <- names(sort(-table(df.combined$BsmtCond)))[1]  
df.combined$BsmtQual[c(2218, 2219)] <- names(sort(-table(df.combined$BsmtQual)))[1]  
  
##Imputing Missing BsmtQual Values  
df.combined$BsmtQual[is.na(df.combined$BsmtQual)] <- 'None'  
df.combined$BsmtQual <- as.integer(revalue(df.combined$BsmtQual, Qualities))

## The following `from` values were not present in `x`: Po

table(df.combined$BsmtQual)

##   
## 0 2 3 4 5   
## 79 88 1285 1209 258

##Imputing Missing BsmtCond Values  
unique(df.combined$BsmtCond)

## [1] "TA" "Gd" NA "Fa" "Po"

df.combined$BsmtCond[is.na(df.combined$BsmtCond)] <- 'None'  
df.combined$BsmtCond <- as.integer(revalue(df.combined$BsmtCond, Qualities))

## The following `from` values were not present in `x`: Ex

table(df.combined$BsmtCond)

##   
## 0 1 2 3 4   
## 79 5 104 2609 122

##Imputing Missing BsmtExposure Values  
unique(df.combined$BsmtExposure)

## [1] "No" "Gd" "Mn" "Av" NA

##Exposure Levels Vector  
Exposure <- c('None' = 0, 'No' = 1, 'Mn' = 2, 'Av' = 3, 'Gd' = 4)  
  
df.combined$BsmtExposure[is.na(df.combined$BsmtExposure)] <- 'None'  
df.combined$BsmtExposure <- as.integer(revalue(df.combined$BsmtExposure, Exposure))  
table(df.combined$BsmtExposure)

##   
## 0 1 2 3 4   
## 79 1907 239 418 276

##Imputing Missing BsmtFinType1 Values  
unique(df.combined$BsmtFinType1)

## [1] "GLQ" "ALQ" "Unf" "Rec" "BLQ" NA "LwQ"

##FinType Levels Vector  
FinType <- c('None'=0, 'Unf'=1, 'LwQ'=2, 'Rec'=3, 'BLQ'=4, 'ALQ'=5, 'GLQ'=6)  
  
df.combined$BsmtFinType1[is.na(df.combined$BsmtFinType1)] <- 'None'  
df.combined$BsmtFinType1<-as.integer(revalue(df.combined$BsmtFinType1, FinType))  
table(df.combined$BsmtFinType1)

##   
## 0 1 2 3 4 5 6   
## 79 851 154 288 269 429 849

##Imputing BsmtFinType2 Values  
unique(df.combined$BsmtFinType2)

## [1] "Unf" "BLQ" NA "ALQ" "Rec" "LwQ" "GLQ"

df.combined$BsmtFinType2[is.na(df.combined$BsmtFinType2)] <- 'None'  
df.combined$BsmtFinType2 <- as.integer(revalue(df.combined$BsmtFinType2, FinType))  
table(df.combined$BsmtFinType2)

##   
## 0 1 2 3 4 5 6   
## 79 2494 87 105 68 52 34

##Identify remaining NA's for Bsmt Values  
df.combined[(is.na(df.combined$BsmtFullBath)|is.na(df.combined$BsmtHalfBath)|is.na(df.combined$BsmtFinSF1)|is.na(df.combined$BsmtFinSF2)|is.na(df.combined$BsmtUnfSF)|is.na(df.combined$TotalBsmtSF)), c('BsmtQual', 'BsmtFullBath', 'BsmtHalfBath', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF')]

## BsmtQual BsmtFullBath BsmtHalfBath BsmtFinSF1 BsmtFinSF2 BsmtUnfSF  
## 2121 0 NA NA NA NA NA  
## 2189 0 NA NA 0 0 0  
## TotalBsmtSF  
## 2121 NA  
## 2189 0

#Imputing Missing BsmtFullBath Values  
unique(df.combined$BsmtFullBath)

## [1] 1 0 2 3 NA

df.combined$BsmtFullBath[is.na(df.combined$BsmtFullBath)] <- 0  
table(df.combined$BsmtFullBath)

##   
## 0 1 2 3   
## 1707 1172 38 2

##Imputing Missing BsmtHalfBath Values  
unique(df.combined$BsmtHalfBath)

## [1] 0 1 2 NA

df.combined$BsmtHalfBath[is.na(df.combined$BsmtHalfBath)] <- 0  
table(df.combined$BsmtHalfBath)

##   
## 0 1 2   
## 2744 171 4

##Imputing Missing BsmtFinSF1 Values  
unique(df.combined$BsmtFinSF1)

## [1] 706 978 486 216 655 732 1369 859 0 851 906 998 737 733  
## [15] 578 646 504 840 188 234 1218 1277 1018 1153 1213 731 643 967  
## [29] 747 280 179 456 1351 24 763 182 104 1810 384 490 649 632  
## [43] 941 739 912 1013 603 1880 565 320 462 228 336 448 1201 33  
## [57] 588 600 713 1046 648 310 1162 520 108 569 1200 224 705 444  
## [71] 250 984 35 774 419 170 1470 938 570 300 120 116 512 567  
## [85] 445 695 405 1005 668 821 432 1300 507 679 1332 209 680 716  
## [99] 1400 416 429 222 57 660 1016 370 351 379 1288 360 639 495  
## [113] 288 1398 477 831 1904 436 352 611 1086 297 626 560 390 566  
## [127] 1126 1036 1088 641 617 662 312 1065 787 468 36 822 378 946  
## [141] 341 16 550 524 56 321 842 689 625 358 402 94 1078 329  
## [155] 929 697 1573 270 922 503 1334 361 672 506 714 403 751 226  
## [169] 620 546 392 421 905 904 430 614 450 210 292 795 1285 819  
## [183] 420 841 281 894 1464 700 262 1274 518 1236 425 692 987 970  
## [197] 28 256 1619 40 846 1124 720 828 1249 810 213 585 129 498  
## [211] 1270 573 1410 1082 236 388 334 874 956 773 399 162 712 609  
## [225] 371 540 72 623 428 350 298 1445 218 985 631 1280 241 690  
## [239] 266 777 812 786 1116 789 1056 50 1128 775 1309 1246 986 616  
## [253] 1518 664 387 471 385 365 1767 133 642 247 331 742 1606 916  
## [267] 185 544 553 326 778 386 426 368 459 1350 1196 630 994 168  
## [281] 1261 1567 299 897 607 836 515 374 1231 111 356 400 698 1247  
## [295] 257 380 27 141 991 650 521 1436 2260 719 377 1330 348 1219  
## [309] 783 969 673 1358 1260 144 584 554 1002 619 180 559 308 866  
## [323] 895 637 604 1302 1071 290 728 2 1441 943 231 414 349 442  
## [337] 328 594 816 1460 1324 1338 685 1422 1283 81 454 903 605 990  
## [351] 206 150 457 48 871 41 674 624 480 1154 738 493 1121 282  
## [365] 500 131 1696 806 1361 920 1721 187 1138 988 193 551 767 1186  
## [379] 892 311 827 543 1003 1059 239 945 20 1455 965 980 863 533  
## [393] 1084 1173 523 1148 191 1234 375 808 724 152 1180 252 832 575  
## [407] 919 439 381 438 549 612 1163 437 394 1416 422 762 975 1097  
## [421] 251 686 656 568 539 862 197 516 663 608 1636 784 249 1040  
## [435] 483 196 572 338 330 156 1390 513 460 659 364 564 306 505  
## [449] 932 750 64 633 1170 899 902 1238 528 1024 1064 285 2188 465  
## [463] 322 860 599 354 63 223 301 443 489 284 294 814 165 552  
## [477] 833 464 936 772 1440 748 982 398 562 484 417 699 696 896  
## [491] 556 1106 651 867 854 1646 1074 536 1172 915 595 1237 273 684  
## [505] 324 1165 138 1513 317 1012 1022 509 900 1085 1104 240 383 644  
## [519] 397 740 837 220 586 535 410 75 824 592 1039 510 423 661  
## [533] 248 704 412 1032 219 708 415 1004 353 702 369 622 212 645  
## [547] 852 1150 1258 275 176 296 538 1157 492 1198 1387 522 658 1216  
## [561] 1480 2096 1159 440 1456 883 547 788 485 340 1220 427 344 756  
## [575] 1540 666 803 1000 885 1386 319 534 125 1314 602 192 593 804  
## [589] 1053 532 1158 1014 194 167 776 5644 694 1572 746 1406 925 482  
## [603] 189 765 80 1443 259 735 734 1447 548 315 1282 408 309 203  
## [617] 865 204 790 1320 769 1070 264 759 1373 976 781 25 1110 404  
## [631] 580 678 958 1336 1079 49 830 923 791 263 935 1051 514 110  
## [645] 1414 126 1129 1298 376 466 244 1137 687 1010 1500 670 944 1188  
## [659] 856 339 481 717 579 274 780 283 474 452 276 960 766 1026  
## [673] 73 736 1319 267 1092 964 954 1346 1433 870 198 1682 238 343  
## [687] 76 615 78 42 469 207 458 476 1341 844 847 850 1965 741  
## [701] 363 225 1333 888 636 726 254 435 389 279 1360 1232 2288 1531  
## [715] 1230 1015 1037 1142 1262 1972 881 876 2146 1557 800 652 494 683  
## [729] 913 1294 2158 682 1430 771 54 52 68 864 140 1733 601 962  
## [743] 1252 121 955 100 1312 172 155 931 872 745 621 433 826 134  
## [757] 169 749 1152 527 342 173 70 1094 820 1021 1359 755 950 606  
## [771] 1259 710 1111 1478 332 793 246 154 65 1476 55 1758 1115 1640  
## [785] 114 718 496 1337 1034 983 1206 890 1023 119 286 1728 1375 1420  
## [799] 2257 1149 1075 372 1204 1073 1087 1660 1096 729 362 537 472 53  
## [813] 764 190 1027 1141 681 813 128 1044 260 583 32 531 148 744  
## [827] 96 590 200 406 175 201 NA 758 221 634 1035 779 1271 355  
## [841] 2085 770 722 1308 688 88 1194 1538 1593 1033 366 1474 1383 893  
## [855] 1029 1223 1011 1571 318 501 785 638 647 838 186 926 1101 1047  
## [869] 797 1558 1328 314 930 725 1151 1304 1812 1684 669 1178 1030 848  
## [883] 918 574 1181 1048 335 1225 727 968 60 937 901 1732 1632 973  
## [897] 910 346 792 654 130 873 908 441 85 242 952 1098 782 122  
## [911] 316 258 587 491 453 557 1080 497 51 502 671 1412 709 132  
## [925] 4010 467 77 113 577 434 1001 1392 1239 924 949 215 1329 1112  
## [939] 796 811 1090 596 1127 205 1191 951 382 373 1505 1290 880 1038  
## [953] 1182 1562 1836 278 181 1118 760 799 996 939 914 271 488 701  
## [967] 455 809 953 208 143 576 347 794 230 261 393 1576 1122 853  
## [981] 475 691 424 305 526 1564 909 1136 1243 149 1224 337

df.combined$BsmtFinSF1[is.na(df.combined$BsmtFinSF1)] <- 0  
table(is.na(df.combined$BsmtFinSF1))

##   
## FALSE   
## 2919

##Imputing Missing BsmtFinSF2 Values  
unique(df.combined$BsmtFinSF2)

## [1] 0 32 668 486 93 491 506 712 362 41 169 869 150 670  
## [15] 28 1080 181 768 215 374 208 441 184 279 306 180 580 690  
## [29] 692 228 125 1063 620 175 820 1474 264 479 147 232 380 544  
## [43] 294 258 121 391 531 344 539 713 210 311 1120 165 532 96  
## [57] 495 174 1127 139 202 645 123 551 219 606 612 480 182 132  
## [71] 336 468 287 35 499 723 119 40 117 239 80 472 64 1057  
## [85] 127 630 128 377 764 345 1085 435 823 500 290 324 634 411  
## [99] 841 1061 466 396 354 149 193 273 465 400 682 557 230 106  
## [113] 791 240 547 469 177 108 600 492 211 168 1031 438 375 144  
## [127] 81 906 608 276 661 68 173 972 105 420 546 334 352 872  
## [141] 110 627 163 1029 78 859 981 42 46 162 350 263 1073 12  
## [155] 159 474 453 684 387 688 252 590 284 622 113 1526 360 774  
## [169] 364 596 884 92 216 136 201 512 247 483 750 60 102 95  
## [183] 63 262 393 286 450 72 243 694 875 507 419 250 116 624  
## [197] 76 270 288 186 449 48 613 852 555 799 811 842 382 456  
## [211] 308 52 196 488 319 NA 956 120 679 604 153 619 6 351  
## [225] 1037 829 38 206 167 543 259 404 138 955 691 66 154 442  
## [239] 448 227 398 722 761 529 522 873 891 755 321 915 417 432  
## [253] 831 278 1020 530 904 156 1393 1039 497 402 748 281 912 373  
## [267] 982 826 850 1164 1083 337 297

df.combined$BsmtFinSF2[is.na(df.combined$BsmtFinSF2)] <- 0  
table(is.na(df.combined$BsmtFinSF2))

##   
## FALSE   
## 2919

##Imputing Missing BsmtUnfSF Values  
unique(df.combined$BsmtUnfSF)

## [1] 150 284 434 540 490 64 317 216 952 140 134 177 175  
## [14] 1494 520 832 426 0 468 525 1158 637 1777 200 204 1566  
## [27] 180 486 207 649 1228 1234 380 408 1117 1097 84 326 445  
## [40] 383 167 465 1296 83 1632 736 192 612 816 32 935 321  
## [53] 860 1410 148 217 530 1346 576 318 1143 1035 440 747 701  
## [66] 343 280 404 840 724 295 1768 448 36 1530 1065 384 1288  
## [79] 684 1013 402 635 163 168 176 370 350 381 410 741 1226  
## [92] 1053 641 516 793 1139 550 905 104 310 252 1125 203 728  
## [105] 732 510 899 1362 30 958 556 413 479 297 658 262 891  
## [118] 1304 519 1907 336 107 432 403 811 396 970 506 884 400  
## [131] 896 253 409 93 1200 572 774 769 1335 340 882 779 112  
## [144] 470 294 1686 360 441 354 700 725 320 554 312 968 504  
## [157] 1107 577 660 99 871 474 289 600 755 625 1121 276 186  
## [170] 1424 1140 375 92 305 1176 78 274 311 710 686 457 1232  
## [183] 1498 1010 160 2336 630 638 162 70 1357 1194 773 483 235  
## [196] 125 1390 594 1694 488 357 626 916 1020 1367 798 452 392  
## [209] 975 361 270 602 1482 680 606 88 342 212 1095 96 628  
## [222] 1560 744 2121 768 386 1468 1145 244 698 1079 570 476 131  
## [235] 184 143 1092 324 1541 1470 536 319 599 622 179 292 286  
## [248] 80 712 291 153 1088 1249 166 906 604 100 818 844 596  
## [261] 210 1603 115 103 673 726 995 967 721 1656 972 460 208  
## [274] 191 438 1869 371 624 552 322 598 268 130 484 785 733  
## [287] 953 847 333 1580 411 982 808 1293 939 784 595 229 114  
## [300] 522 735 405 117 961 1286 672 1141 806 165 1064 1063 245  
## [313] 1276 892 1008 499 1316 463 242 444 281 35 356 988 580  
## [326] 651 619 544 387 901 926 135 648 75 788 1307 1078 1258  
## [339] 273 1436 557 930 780 813 878 122 248 588 524 288 389  
## [352] 424 1375 1626 406 298 2153 417 739 225 611 237 290 264  
## [365] 238 363 190 1969 697 414 316 466 420 254 960 397 1191  
## [378] 548 50 178 1368 169 748 689 1264 467 605 1257 551 678  
## [391] 707 880 378 223 578 969 379 765 149 912 620 1709 132  
## [404] 993 197 1374 90 195 706 1163 367 1122 1515 55 1497 450  
## [417] 846 23 390 861 285 1050 331 2042 1237 113 742 924 512  
## [430] 119 314 308 293 537 126 427 309 914 173 1774 823 485  
## [443] 1116 978 636 564 108 1184 796 366 300 542 645 664 756  
## [456] 247 776 849 1392 38 1406 111 545 121 2046 161 261 567  
## [469] 1195 874 1342 151 989 1073 927 219 224 526 1164 761 461  
## [482] 876 859 171 718 138 941 464 250 72 508 1584 415 82  
## [495] 948 893 864 1349 76 487 652 1240 801 279 1030 348 234  
## [508] 1198 740 89 586 323 1836 480 456 1935 338 1594 102 374  
## [521] 1413 491 1129 255 1496 650 1926 154 999 1734 124 1417 15  
## [534] 834 1649 936 778 1489 442 1434 352 458 1221 1099 416 1800  
## [547] 227 907 528 189 1273 563 372 702 1090 435 198 1372 174  
## [560] 1638 894 299 105 676 1120 431 218 110 795 1098 1043 481  
## [573] 666 142 447 783 1670 277 412 794 239 662 1072 717 546  
## [586] 430 422 188 266 1181 1753 964 1450 1905 1480 772 1032 220  
## [599] 187 29 495 640 193 196 720 918 1428 77 1266 1128 692  
## [612] 770 750 1442 1007 501 691 1550 1680 1330 1710 746 814 515  
## [625] 571 359 355 301 668 920 1055 1420 1752 304 1302 833 133  
## [638] 549 705 722 799 462 429 810 155 170 230 1459 1082 758  
## [651] 1290 1074 251 172 868 797 365 418 730 533 671 1012 1528  
## [664] 1005 1373 500 762 752 399 1042 40 26 932 278 459 568  
## [677] 1502 543 574 977 449 983 731 120 538 831 994 341 879  
## [690] 815 1212 866 1630 328 141 364 1380 81 303 940 764 1048  
## [703] 334 1689 690 792 585 473 246 1045 1405 201 14 841 1104  
## [716] 241 925 2002 74 661 708 1152 256 804 812 1085 344 425  
## [729] 1616 976 496 349 971 1393 1622 1352 1795 1017 1588 428 803  
## [742] 693 858 1284 1203 1652 39 539 1217 257 715 616 240 315  
## [755] 1351 1026 1571 156 61 95 482 1094 60 862 221 791 398  
## [768] 777 503 734 709 1252 656 1319 1422 560 1573 589 877 136  
## [781] 137 763 233 789 663 327 836 1590 1544 1794 1473 1093 1324  
## [794] 58 1629 1595 1218 54 610 659 1323 534 228 1604 827 346  
## [807] 455 634 144 164 888 232 1250 86 296 1040 974 618 850  
## [820] 657 1115 1958 1214 1430 1344 395 1216 388 590 98 158 243  
## [833] 1077 1058 471 1180 79 498 1736 632 1598 643 1084 1451 745  
## [846] 1204 282 1696 1614 1402 1348 222 1087 835 1114 1332 345 28  
## [859] 182 1376 1726 183 249 306 1619 1568 265 613 507 313 565  
## [872] 727 1866 1054 1313 1248 996 206 559 1041 339 269 677 917  
## [885] 1022 679 938 851 46 825 382 1474 704 513 335 1068 944  
## [898] 608 615 1678 94 583 674 621 91 561 118 581 760 1527  
## [911] 898 1280 332 1018 547 1488 469 738 1444 213 25 275 575  
## [924] 908 454 1765 1486 1347 1318 1146 1173 1519 1242 1341 226 1339  
## [937] 1660 1162 439 1328 1211 194 587 147 584 1439 723 1625 1728  
## [950] 505 529 437 330 998 1824 675 453 181 828 931 407 869  
## [963] 329 949 475 992 497 NA 897 1046 1272 647 451 852 639  
## [976] 1615 631 782 1421 1508 1327 541 591 555 821 592 910 1495  
## [989] 443 873 393 325 271 749 52 1168 903 1559 53 1028  
## [ reached getOption("max.print") -- omitted 136 entries ]

df.combined$BsmtUnfSF[is.na(df.combined$BsmtUnfSF)] <- 0  
table(is.na(df.combined$BsmtUnfSF))

##   
## FALSE   
## 2919

##Imputing Missing TotalBsmtSF Values  
unique(df.combined$TotalBsmtSF)

## [1] 856 1262 920 756 1145 796 1686 1107 952 991 1040 1175 912  
## [14] 1494 1253 832 1004 0 1114 1029 1158 637 1777 1060 1566 900  
## [27] 1704 1484 520 649 1228 1234 1398 1561 1117 1097 1297 1057 1088  
## [40] 1350 840 938 1150 1752 1434 1656 736 955 794 816 1842 384  
## [53] 1425 970 860 1410 780 530 1370 576 1143 1947 1453 747 1304  
## [66] 2223 845 1086 462 672 1768 440 896 1237 1563 1065 1288 684  
## [79] 612 1013 990 1235 876 1214 824 680 1588 960 458 950 1610  
## [92] 741 1226 1053 641 789 793 1844 994 1264 1809 1028 729 1092  
## [105] 1125 1673 728 732 1080 1199 1362 1078 660 1008 924 992 1063  
## [118] 1267 1461 1907 928 864 1734 910 1490 1728 715 884 969 1710  
## [131] 825 1602 1200 572 774 1392 1232 1572 1541 882 1149 644 1617  
## [144] 1582 720 1064 1606 1202 1151 1052 2216 968 504 1188 1593 853  
## [157] 725 1431 855 1726 1360 755 1713 1121 1196 617 848 1424 1140  
## [170] 1100 1157 1212 689 1070 1436 686 798 1248 1498 1010 713 2392  
## [183] 630 1203 483 1373 1194 1462 894 1414 996 1694 735 540 626  
## [196] 948 1845 1020 1367 1444 1573 1302 1314 975 1604 963 1482 506  
## [209] 926 1422 802 740 1095 1385 1152 1240 1560 2121 1160 807 1468  
## [222] 1575 625 858 698 1079 768 795 1416 1003 702 1165 1470 2000  
## [235] 700 319 861 1896 697 972 2136 716 1347 1372 1249 1136 1502  
## [248] 1162 710 1719 1383 844 596 1056 3206 1358 943 1499 1922 1536  
## [261] 1208 1215 967 721 1684 536 958 1478 764 1848 1869 616 624  
## [274] 940 1142 1062 888 883 1394 1099 1268 953 744 608 847 683  
## [287] 870 1580 1856 982 1026 1293 939 784 1256 658 1041 1682 804  
## [300] 788 1144 961 1260 1310 1141 806 1281 1034 1276 1340 1344 988  
## [313] 651 1518 907 901 765 799 648 3094 1440 1258 915 1517 930  
## [326] 813 1533 872 1242 1364 588 709 560 1375 1277 1626 1488 808  
## [339] 547 1976 2153 1705 1833 1792 1216 999 1113 1073 954 264 1269  
## [352] 190 3200 866 1501 777 1218 1368 1084 2006 1244 3138 1379 1257  
## [365] 1452 528 2035 611 707 880 1051 1581 1838 1650 723 654 1204  
## [378] 1069 1709 998 993 1374 1389 1163 1122 1496 846 372 1164 1050  
## [391] 2042 1868 1437 742 770 1722 1814 1430 1058 908 600 965 1032  
## [404] 1299 1120 936 783 1822 1522 980 1116 978 1156 636 1554 1386  
## [417] 811 1520 1952 1766 981 1094 2109 525 776 1486 1629 1138 2077  
## [430] 1406 1021 1408 738 1477 2046 923 1291 1195 1190 874 551 1419  
## [443] 2444 1210 927 1112 1391 1800 360 1473 1643 1324 270 859 718  
## [456] 1176 1311 971 1742 941 1698 1584 1595 868 1153 893 1349 1337  
## [469] 1720 1479 1030 1318 1252 983 1860 836 1935 1614 761 1413 956  
## [482] 712 650 773 1926 731 1417 1024 849 1442 1649 1568 778 1489  
## [495] 2078 1454 1516 1067 1559 1127 1390 1273 918 1763 1090 1054 1039  
## [508] 1148 1002 1638 105 676 1184 1109 892 2217 1505 1059 951 2330  
## [521] 1670 1623 1017 1105 1001 546 480 1134 1104 1272 1316 1126 1181  
## [534] 1753 964 1466 925 1905 1500 585 1632 819 1616 1161 828 945  
## [547] 979 561 696 1330 817 1098 1428 673 1241 944 1225 1266 1128  
## [560] 485 1930 1396 916 822 750 1700 1007 1187 691 1574 1680 1346  
## [573] 985 1657 602 1022 1082 810 1504 1220 1132 1565 1338 1654 1620  
## [586] 1055 800 1306 1475 2524 1992 1193 973 854 662 1103 1154 942  
## [599] 1048 727 690 1096 1459 1251 1247 1074 1271 290 655 1463 1836  
## [612] 803 833 408 533 1012 1552 1005 1530 974 1567 1006 1042 1298  
## [625] 704 932 1219 1296 1198 959 1261 1598 1683 818 1600 2396 1624  
## [638] 831 1224 663 879 815 1630 2158 931 1660 559 1300 1702 1075  
## [651] 1361 1106 1476 1689 2076 792 2110 1405 1192 746 1986 841 2002  
## [664] 1332 935 1019 661 1309 1328 1085 6110 1246 771 976 1652 1278  
## [677] 1902 1274 1393 1622 1352 420 1795 544 1510 911 693 1284 1732  
## [690] 2033 570 1980 814 873 757 1108 2633 1571 984 1205 714 1746  
## [703] 1525 482 1356 862 839 1286 1485 1594 622 791 708 1223 913  
## [716] 656 1319 1932 539 1221 1542 1329 1280 763 1168 1590 1544 2846  
## [729] 1671 1231 1642 1492 1829 1209 782 1480 1206 1395 827 1027 678  
## [742] 346 835 1124 1832 1420 1172 1508 1250 1433 946 1222 878 1978  
## [755] 381 1528 423 1191 629 1049 1243 1958 1336 1068 1679 2208 1418  
## [768] 1587 1427 1043 1621 1180 1776 1365 2020 2630 1736 1782 1739 1774  
## [781] 1760 2452 2492 2200 1884 1451 1712 745 1177 1455 2024 1173 1696  
## [794] 2458 1402 1553 812 1512 1450 754 2014 1376 1740 392 752 1313  
## [807] 1292 1562 1169 1382 1866 631 1031 699 405 416 1380 160 1045  
## [820] 240 801 677 917 245 297 468 1072 456 552 1089 1555 1282  
## [833] 1254 1449 914 1569 1578 1014 1678 583 516 1803 760 1596 902  
## [846] 957 2190 1641 1645 1348 1664 1675 2048 1211 1236 2418 1950 1850  
## [859] 2535 1603 1765 1858 1342 1415 1146 1037 1519 1982 1341 1460 1363  
## [872] 886 890 1557 1230 1426 1625 1118 348 1036 1824 1312 1081 192  
## [885] 481 904 407 448 797 869 554 949 739 565 NA 989 451  
## [898] 1046 450 1524 1509 621 1077 628 1615 2271 1751 1401 1182 1076  
## [911] 1357 1778 1129 850 1445 1564 1351 1091 1898 173 356 592 1840  
## [924] 352 1495 1432 1666 1964 1189 550 526 903 1166 2660 1612 2220  
## [937] 1529 1259 1061 1790 2108 1934 1994 2552 2320 1802 1706 1317 895  
## [950] 1721 1577 1326 1779 1066 1325 1369 1966 1538 1335 1685 1044 1378  
## [963] 1511 1550 1339 2461 1295 1093 1174 1130 1592 666 370 1331 922  
## [976] 1170 1377 687 966 671 409 484 1038 779 2140 1546 1270 929  
## [989] 635 1015 995 385 1556 1531 1179 734 5095 1290 531 851  
## [ reached getOption("max.print") -- omitted 59 entries ]

df.combined$TotalBsmtSF[is.na(df.combined$TotalBsmtSF)] <- 0  
table(is.na(df.combined$TotalBsmtSF))

##   
## FALSE   
## 2919

##Imputing Missing Masonry Values  
length(which(is.na(df.combined$MasVnrType) & is.na(df.combined$MasVnrArea)))

## [1] 23

##Find Missing MasVrnType  
df.combined[is.na(df.combined$MasVnrType) & !is.na(df.combined$MasVnrArea), c('MasVnrType', 'MasVnrArea')]

## MasVnrType MasVnrArea  
## 2611 <NA> 198

##Impute #2611 Missing MasVrnType with the mode  
df.combined$MasVnrType[2611] <- names(sort(-table(df.combined$MasVnrType)))[2]  
df.combined[2611, c('MasVnrType', 'MasVnrArea')]

## MasVnrType MasVnrArea  
## 2611 BrkFace 198

##Impute Missing MasVnrType Values  
unique(df.combined$MasVnrType)

## [1] "BrkFace" "None" "Stone" "BrkCmn" NA

df.combined$MasVnrType[is.na(df.combined$MasVnrType)] <- 'None'  
table(is.na(df.combined$MasVnrType))

##   
## FALSE   
## 2919

##MasVnrType by Median SalePrice  
df.combined[!is.na(df.combined$SalePrice),] %>%   
 group\_by(MasVnrType) %>%   
 summarise(median = median(SalePrice), counts = n()) %>%   
 arrange(median)

## # A tibble: 4 x 3  
## MasVnrType median counts  
## <chr> <dbl> <int>  
## 1 BrkCmn 139000 15  
## 2 None 143125 872  
## 3 BrkFace 181000 445  
## 4 Stone 246839 128

##Masonry Levels Vector  
Masonry <- c('None' = 0, 'BrkCmn' = 0, 'BrkFace' = 1, 'Stone' = 2)  
  
df.combined$MasVnrType <- as.integer(revalue(df.combined$MasVnrType, Masonry))  
table(df.combined$MasVnrType)

##   
## 0 1 2   
## 1790 880 249

##Imputing Missing MasVnrArea Values  
unique(df.combined$MasVnrArea)

## [1] 196 0 162 350 186 240 286 306 212 180 380 281 640 200  
## [15] 246 132 650 101 412 272 456 1031 178 573 344 287 167 1115  
## [29] 40 104 576 443 468 66 22 284 76 203 68 183 48 28  
## [43] 336 600 768 480 220 184 1129 116 135 266 85 309 136 288  
## [57] 70 320 50 120 436 252 84 664 226 300 653 112 491 268  
## [71] 748 98 275 138 205 262 128 260 153 64 312 16 922 142  
## [85] 290 127 506 297 NA 604 254 36 102 472 481 108 302 172  
## [99] 399 270 46 210 174 348 315 299 340 166 72 31 34 238  
## [113] 1600 365 56 150 278 256 225 370 388 175 296 146 113 176  
## [127] 616 30 106 870 362 530 500 510 247 305 255 125 100 432  
## [141] 126 473 74 145 232 376 42 161 110 18 224 248 80 304  
## [155] 215 772 435 378 562 168 89 285 360 94 333 921 762 594  
## [169] 219 188 479 584 182 250 292 245 207 82 97 335 208 420  
## [183] 170 459 280 99 192 204 233 156 452 513 261 164 259 209  
## [197] 263 216 351 660 381 54 528 258 464 57 147 1170 293 630  
## [211] 466 109 41 160 289 651 169 95 442 202 338 894 328 673  
## [225] 603 1 375 90 38 157 11 140 130 148 860 424 1047 243  
## [239] 816 387 223 158 137 115 189 274 117 60 122 92 415 760  
## [253] 27 75 361 105 342 298 541 236 144 423 44 151 975 450  
## [267] 230 571 24 53 206 14 324 295 396 67 154 425 45 1378  
## [281] 337 149 143 51 171 234 63 766 32 81 163 554 218 632  
## [295] 114 567 359 451 621 788 86 796 391 228 88 165 428 410  
## [309] 564 368 318 579 65 705 408 244 123 366 731 448 294 310  
## [323] 237 426 96 438 194 119 20 504 492 615 1095 1159 265 91  
## [337] 771 47 177 371 430 440 229 726 418 724 383 730 470 308  
## [351] 634 372 198 121 264 141 283 509 217 3 657 124 444 23  
## [365] 242 364 352 406 402 422 356 680 1110 221 714 647 1290 495  
## [379] 568 179 1050 187 52 276 39 190 251 227 134 222 58 668  
## [393] 674 197 710 945 549 253 400 970 502 394 235 515 526 754  
## [407] 353 525 87 291 69 279 323 214 519 1224 652 886 902 434  
## [421] 662 734 550 514 385 518 572 322 877 397 738 501 118 692  
## [435] 332 522 379 532 62 199 355 405 327 257 382

df.combined$MasVnrArea[is.na(df.combined$MasVnrArea)] <- 0  
table(is.na(df.combined$MasVnrArea))

##   
## FALSE   
## 2919

##Imputing Missing MSZoning Values  
unique(df.combined$MSZoning)

## [1] "RL" "RM" "C (all)" "FV" "RH" NA

table(is.na(df.combined$MSZoning))

##   
## FALSE TRUE   
## 2915 4

##Imputing Mode of MSZoning Values  
df.combined$MSZoning[is.na(df.combined$MSZoning)] <- names(sort(-table(df.combined$MSZoning)))[1]  
df.combined$MSZoning <- as.factor(df.combined$MSZoning)  
  
table(df.combined$MSZoning)

##   
## C (all) FV RH RL RM   
## 25 139 26 2269 460

sum(table(df.combined$MSZoning))

## [1] 2919

##Imputing Missing Kitchen Values  
unique(df.combined$KitchenQual)

## [1] "Gd" "TA" "Ex" "Fa" NA

table(is.na(df.combined$KitchenQual))

##   
## FALSE TRUE   
## 2918 1

##Imputing mode of KitchenQual Values  
df.combined$KitchenQual[is.na(df.combined$KitchenQual)]

## [1] NA

df.combined$KitchenQual[is.na(df.combined$KitchenQual)] <- names(sort(-table(df.combined$KitchenQual)))[1]  
df.combined$KitchenQual<-as.integer(revalue(df.combined$KitchenQual, Qualities))

## The following `from` values were not present in `x`: None, Po

table(df.combined$KitchenQual)

##   
## 2 3 4 5   
## 70 1493 1151 205

sum(table(df.combined$KitchenQual))

## [1] 2919

##Verifying KitchenAbvGr Values  
unique(df.combined$KitchenAbvGr)

## [1] 1 2 3 0

table(is.na(df.combined$KitchenAbvGr))

##   
## FALSE   
## 2919

table(df.combined$KitchenAbvGr)

##   
## 0 1 2 3   
## 3 2785 129 2

sum(table(df.combined$KitchenAbvGr))

## [1] 2919

##Imputing Missing Utilities Values  
unique(df.combined$Utilities)

## [1] "AllPub" "NoSeWa" NA

table(is.na(df.combined$Utilities))

##   
## FALSE TRUE   
## 2917 2

##Only 1 House does not have public ultilities(in training set), therefore variable is useless for prediction  
kable(df.combined[is.na(df.combined$Utilities) | df.combined$Utilities == 'NoSeWa', 1:9])

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | MSSubClass | MSZoning | LotFrontage | LotArea | Street | Alley | LotShape | LandContour | Utilities |
| 945 | 20 | RL | 82 | 14375 | Pave | None | 2 | Lvl | NoSeWa |
| 1916 | 30 | RL | 109 | 21780 | Grvl | None | 3 | Lvl | NA |
| 1946 | 20 | RL | 64 | 31220 | Pave | None | 2 | Bnk | NA |

df.combined$Utilities <- NULL

##Imputing Home Functionality Values  
unique(df.combined$Functional)

## [1] "Typ" "Min1" "Maj1" "Min2" "Mod" "Maj2" "Sev" NA

table(is.na(df.combined$Functional))

##   
## FALSE TRUE   
## 2917 2

##Impute mode of Functional Values  
df.combined$Functional[is.na(df.combined$Functional)] <- names(sort(-table(df.combined$Functional)))[1]  
##Functional Levels Vector  
Functional <- c('Sal' = 0, 'Sev' = 1, 'Maj2' = 2, 'Maj1' = 3, 'Mod' = 4, 'Min2' = 5, 'Min1' = 6, 'Typ' = 7)  
  
df.combined$Functional <- as.integer(revalue(df.combined$Functional, Functional))

## The following `from` values were not present in `x`: Sal

table(df.combined$Functional)

##   
## 1 2 3 4 5 6 7   
## 2 9 19 35 70 65 2719

sum(table(df.combined$Functional))

## [1] 2919

##Imputing Exterior Values  
unique(df.combined$Exterior1st)

## [1] "VinylSd" "MetalSd" "Wd Sdng" "HdBoard" "BrkFace" "WdShing" "CemntBd"  
## [8] "Plywood" "AsbShng" "Stucco" "BrkComm" "AsphShn" "Stone" "ImStucc"  
## [15] "CBlock" NA

table(is.na(df.combined$Exterior1st))

##   
## FALSE TRUE   
## 2918 1

##Imputing Mode of Exterior1st Values  
df.combined$Exterior1st[is.na(df.combined$Exterior1st)] <- names(sort(-table(df.combined$Exterior1st)))[1]  
df.combined$Exterior1st <- as.factor(df.combined$Exterior1st)  
table(df.combined$Exterior1st)

##   
## AsbShng AsphShn BrkComm BrkFace CBlock CemntBd HdBoard ImStucc MetalSd   
## 44 2 6 87 2 126 442 1 450   
## Plywood Stone Stucco VinylSd Wd Sdng WdShing   
## 221 2 43 1026 411 56

sum(table(df.combined$Exterior1st))

## [1] 2919

##Imputing Exterior2nd Values  
unique(df.combined$Exterior2nd)

## [1] "VinylSd" "MetalSd" "Wd Shng" "HdBoard" "Plywood" "Wd Sdng" "CmentBd"  
## [8] "BrkFace" "Stucco" "AsbShng" "Brk Cmn" "ImStucc" "AsphShn" "Stone"   
## [15] "Other" "CBlock" NA

table(is.na(df.combined$Exterior2nd))

##   
## FALSE TRUE   
## 2918 1

##Imputing Mode of Exterior1st Values  
df.combined$Exterior2nd[is.na(df.combined$Exterior2nd)] <- names(sort(-table(df.combined$Exterior2nd)))[1]  
df.combined$Exterior2nd <- as.factor(df.combined$Exterior2nd)  
table(df.combined$Exterior2nd)

##   
## AsbShng AsphShn Brk Cmn BrkFace CBlock CmentBd HdBoard ImStucc MetalSd   
## 38 4 22 47 3 126 406 15 447   
## Other Plywood Stone Stucco VinylSd Wd Sdng Wd Shng   
## 1 270 6 47 1015 391 81

sum(table(df.combined$Exterior2nd))

## [1] 2919

##Imputing ExterQual Values  
unique(df.combined$ExterQual)

## [1] "Gd" "TA" "Ex" "Fa"

table(is.na(df.combined$ExterQual))

##   
## FALSE   
## 2919

##Imputing Quality Levels Vector  
df.combined$ExterQual <- as.integer(revalue(df.combined$ExterQual, Qualities))

## The following `from` values were not present in `x`: None, Po

table(df.combined$ExterQual)

##   
## 2 3 4 5   
## 35 1798 979 107

sum(table(df.combined$ExterQual))

## [1] 2919

##Imputing ExterCond Values  
unique(df.combined$ExterCond)

## [1] "TA" "Gd" "Fa" "Po" "Ex"

table(is.na(df.combined$ExterCond))

##   
## FALSE   
## 2919

##Imputing Quality Levels Vector  
df.combined$ExterCond <- as.integer(revalue(df.combined$ExterCond, Qualities))

## The following `from` values were not present in `x`: None

table(df.combined$ExterCond)

##   
## 1 2 3 4 5   
## 3 67 2538 299 12

sum(table(df.combined$ExterCond))

## [1] 2919

##Imputing Electrical System Values  
unique(df.combined$Electrical)

## [1] "SBrkr" "FuseF" "FuseA" "FuseP" "Mix" NA

table(is.na(df.combined$Electrical))

##   
## FALSE TRUE   
## 2918 1

##Imputing Mode of Electrical Values  
df.combined$Electrical[is.na(df.combined$Electrical)] <- names(sort(-table(df.combined$Electrical)))[1]  
df.combined$Electrical <- as.factor(df.combined$Electrical)  
table(df.combined$Electrical)

##   
## FuseA FuseF FuseP Mix SBrkr   
## 188 50 8 1 2672

sum(table(df.combined$Electrical))

## [1] 2919

##Imputing Missing Sale Type & Condition Values  
unique(df.combined$SaleType)

## [1] "WD" "New" "COD" "ConLD" "ConLI" "CWD" "ConLw" "Con"   
## [9] "Oth" NA

table(is.na(df.combined$SaleType))

##   
## FALSE TRUE   
## 2918 1

##Imputing Mode of SaleType Values  
df.combined$SaleType[is.na(df.combined$SaleType)] <- names(sort(-table(df.combined$SaleType)))[1]  
df.combined$SaleType <- as.factor(df.combined$SaleType)  
table(df.combined$SaleType)

##   
## COD Con ConLD ConLI ConLw CWD New Oth WD   
## 87 5 26 9 8 12 239 7 2526

sum(table(df.combined$SaleType))

## [1] 2919

##Imputing SaleCondition Values  
unique(df.combined$SaleCondition)

## [1] "Normal" "Abnorml" "Partial" "AdjLand" "Alloca" "Family"

table(is.na(df.combined$SaleCondition))

##   
## FALSE   
## 2919

df.combined$SaleCondition <- as.factor(df.combined$SaleCondition)  
table(df.combined$SaleCondition)

##   
## Abnorml AdjLand Alloca Family Normal Partial   
## 190 12 24 46 2402 245

sum(table(df.combined$SaleCondition))

## [1] 2919

##Impute Pavement of Street & Driveway Values  
unique(df.combined$Street)

## [1] "Pave" "Grvl"

table(is.na(df.combined$Street))

##   
## FALSE   
## 2919

##Street Levels Vector  
StreetLevels <- c('Grvl' = 0, 'Pave' = 1)  
  
df.combined$Street <- as.integer(revalue(df.combined$Street, StreetLevels))  
table(df.combined$Street)

##   
## 0 1   
## 12 2907

sum(table(df.combined$Street))

## [1] 2919

##Impute PavedDrive Values  
unique(df.combined$PavedDrive)

## [1] "Y" "N" "P"

table(is.na(df.combined$PavedDrive))

##   
## FALSE   
## 2919

##PavedDrive Levels Vector  
PavedDrive <- c('N' = 0, 'P' = 1, 'Y' = 2)  
  
df.combined$PavedDrive <- as.integer(revalue(df.combined$PavedDrive, PavedDrive))  
table(df.combined$PavedDrive)

##   
## 0 1 2   
## 216 62 2641

sum(table(df.combined$PavedDrive))

## [1] 2919

##Factorizing remaining Categoric Variables that are without NA's  
character.VarNames <- names(df.combined[, sapply(df.combined, is.character)])  
character.VarNames

## [1] "LandContour" "LandSlope" "Neighborhood" "Condition1"   
## [5] "Condition2" "BldgType" "HouseStyle" "RoofStyle"   
## [9] "RoofMatl" "Foundation" "Heating" "HeatingQC"   
## [13] "CentralAir"

cat(length(character.VarNames), 'remaining variables with character values')

## 13 remaining variables with character values

##Factorize Foundation Values  
unique(df.combined$Foundation)

## [1] "PConc" "CBlock" "BrkTil" "Wood" "Slab" "Stone"

table(is.na(df.combined$Foundation))

##   
## FALSE   
## 2919

df.combined$Foundation <- as.factor(df.combined$Foundation)  
table(df.combined$Foundation)

##   
## BrkTil CBlock PConc Slab Stone Wood   
## 311 1235 1308 49 11 5

sum(table(df.combined$Foundation))

## [1] 2919

##Factorize Heating & Air Values  
unique(df.combined$Heating)

## [1] "GasA" "GasW" "Grav" "Wall" "OthW" "Floor"

table(is.na(df.combined$Heating))

##   
## FALSE   
## 2919

df.combined$Heating <- as.factor(df.combined$Heating)  
table(df.combined$Heating)

##   
## Floor GasA GasW Grav OthW Wall   
## 1 2874 27 9 2 6

sum(table(df.combined$Heating))

## [1] 2919

##Ordinalize Heating QC Values  
unique(df.combined$HeatingQC)

## [1] "Ex" "Gd" "TA" "Fa" "Po"

table(is.na(df.combined$HeatingQC))

##   
## FALSE   
## 2919

df.combined$HeatingQC <- as.integer(revalue(df.combined$HeatingQC, Qualities))

## The following `from` values were not present in `x`: None

table(df.combined$HeatingQC)

##   
## 1 2 3 4 5   
## 3 92 857 474 1493

sum(table(df.combined$HeatingQC))

## [1] 2919

##Factorize CentralAir Values  
unique(df.combined$CentralAir)

## [1] "Y" "N"

table(is.na(df.combined$CentralAir))

##   
## FALSE   
## 2919

df.combined$CentralAir <- as.integer(revalue(df.combined$CentralAir, c('N' = 0, 'Y' = 1)))  
table(df.combined$CentralAir)

##   
## 0 1   
## 196 2723

sum(table(df.combined$CentralAir))

## [1] 2919

##Factorize Roof Values  
unique(df.combined$RoofStyle)

## [1] "Gable" "Hip" "Gambrel" "Mansard" "Flat" "Shed"

table(is.na(df.combined$RoofStyle))

##   
## FALSE   
## 2919

df.combined$RoofStyle <- as.factor(df.combined$RoofStyle)  
table(df.combined$RoofStyle)

##   
## Flat Gable Gambrel Hip Mansard Shed   
## 20 2310 22 551 11 5

sum(table(df.combined$RoofStyle))

## [1] 2919

##Factorize RoofMatl  
unique(df.combined$RoofMatl)

## [1] "CompShg" "WdShngl" "Metal" "WdShake" "Membran" "Tar&Grv" "Roll"   
## [8] "ClyTile"

table(is.na(df.combined$RoofMatl))

##   
## FALSE   
## 2919

df.combined$RoofMatl <- as.factor(df.combined$RoofMatl)  
table(df.combined$RoofMatl)

##   
## ClyTile CompShg Membran Metal Roll Tar&Grv WdShake WdShngl   
## 1 2876 1 1 1 23 9 7

sum(table(df.combined$RoofMatl))

## [1] 2919

##Factorize Land Values  
unique(df.combined$LandContour)

## [1] "Lvl" "Bnk" "Low" "HLS"

table(is.na(df.combined$LandContour))

##   
## FALSE   
## 2919

df.combined$LandContour <- as.factor(df.combined$LandContour)  
table(df.combined$LandContour)

##   
## Bnk HLS Low Lvl   
## 117 120 60 2622

sum(table(df.combined$LandContour))

## [1] 2919

##Label Encode LandSlope Values  
unique(df.combined$LandSlope)

## [1] "Gtl" "Mod" "Sev"

table(is.na(df.combined$LandSlope))

##   
## FALSE   
## 2919

##LandSlope Levels Vector  
LandSlope <- c('Sev' = 0, 'Mod' = 1, 'Gtl' = 2)  
  
df.combined$LandSlope <- as.integer(revalue(df.combined$LandSlope, LandSlope))  
table(df.combined$LandSlope)

##   
## 0 1 2   
## 16 125 2778

sum(table(df.combined$LandSlope))

## [1] 2919

##Factorize Dwelling Values  
unique(df.combined$BldgType)

## [1] "1Fam" "2fmCon" "Duplex" "TwnhsE" "Twnhs"

table(is.na(df.combined$BldgType))

##   
## FALSE   
## 2919

df.combined$BldgType <- as.factor(df.combined$BldgType)  
table(df.combined$BldgType)

##   
## 1Fam 2fmCon Duplex Twnhs TwnhsE   
## 2425 62 109 96 227

sum(table(df.combined$BldgType))

## [1] 2919

##Factorize HouseStyle Values  
unique(df.combined$HouseStyle)

## [1] "2Story" "1Story" "1.5Fin" "1.5Unf" "SFoyer" "SLvl" "2.5Unf" "2.5Fin"

table(is.na(df.combined$HouseStyle))

##   
## FALSE   
## 2919

df.combined$HouseStyle <- as.factor(df.combined$HouseStyle)  
table(df.combined$HouseStyle)

##   
## 1.5Fin 1.5Unf 1Story 2.5Fin 2.5Unf 2Story SFoyer SLvl   
## 314 19 1471 8 24 872 83 128

sum(table(df.combined$HouseStyle))

## [1] 2919

##Factorize Neighborhood & Condition Values  
unique(df.combined$Neighborhood)

## [1] "CollgCr" "Veenker" "Crawfor" "NoRidge" "Mitchel" "Somerst" "NWAmes"   
## [8] "OldTown" "BrkSide" "Sawyer" "NridgHt" "NAmes" "SawyerW" "IDOTRR"   
## [15] "MeadowV" "Edwards" "Timber" "Gilbert" "StoneBr" "ClearCr" "NPkVill"  
## [22] "Blmngtn" "BrDale" "SWISU" "Blueste"

table(is.na(df.combined$Neighborhood))

##   
## FALSE   
## 2919

df.combined$Neighborhood <- as.factor(df.combined$Neighborhood)  
table(df.combined$Neighborhood)

##   
## Blmngtn Blueste BrDale BrkSide ClearCr CollgCr Crawfor Edwards Gilbert   
## 28 10 30 108 44 267 103 194 165   
## IDOTRR MeadowV Mitchel NAmes NoRidge NPkVill NridgHt NWAmes OldTown   
## 93 37 114 443 71 23 166 131 239   
## Sawyer SawyerW Somerst StoneBr SWISU Timber Veenker   
## 151 125 182 51 48 72 24

sum(table(df.combined$Neighborhood))

## [1] 2919

##Factorize Condition1 Values  
unique(df.combined$Condition1)

## [1] "Norm" "Feedr" "PosN" "Artery" "RRAe" "RRNn" "RRAn" "PosA"   
## [9] "RRNe"

table(is.na(df.combined$Condition1))

##   
## FALSE   
## 2919

df.combined$Condition1 <- as.factor(df.combined$Condition1)  
table(df.combined$Condition1)

##   
## Artery Feedr Norm PosA PosN RRAe RRAn RRNe RRNn   
## 92 164 2511 20 39 28 50 6 9

sum(table(df.combined$Condition1))

## [1] 2919

##Factorize Condition2 Values  
unique(df.combined$Condition2)

## [1] "Norm" "Artery" "RRNn" "Feedr" "PosN" "PosA" "RRAn" "RRAe"

table(is.na(df.combined$Condition2))

##   
## FALSE   
## 2919

df.combined$Condition2 <- as.factor(df.combined$Condition2)  
table(df.combined$Condition2)

##   
## Artery Feedr Norm PosA PosN RRAe RRAn RRNn   
## 5 13 2889 4 4 1 1 2

sum(table(df.combined$Condition2))

## [1] 2919

str(df.combined$YrSold)

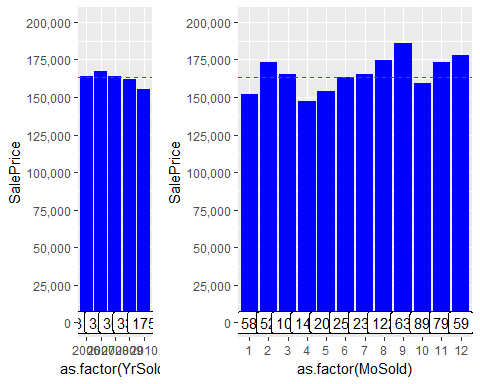
## int [1:2919] 2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...

##Factorize MoSold  
str(df.combined$MoSold)

## int [1:2919] 2 5 9 2 12 10 8 11 4 1 ...

df.combined$MoSold <- as.factor(df.combined$MoSold)

##SalePrice vs. YrSold EDA (Dashed line is median SalePrice)  
year.sold <- ggplot(df.combined[!is.na(df.combined$SalePrice),], aes(x = as.factor(YrSold), y = SalePrice)) +  
 geom\_bar(stat = 'summary', fun.y = "median", fill = 'blue') +  
 scale\_y\_continuous(breaks = seq(0, 800000, by = 25000), labels = scales::comma) +  
 geom\_label(stat = "count", aes(label = ..count.., y = ..count..)) +  
 coord\_cartesian(ylim = c(0, 200000)) +  
 geom\_hline(yintercept = 163000, linetype = "dashed", color = "red")  
  
##SalePrice vs. MoSold EDA (Dashed line is median SalePrice)  
month.sold <- ggplot(df.combined[!is.na(df.combined$SalePrice),], aes(x = as.factor(MoSold), y = SalePrice)) +  
 geom\_bar(stat = 'summary', fun.y = "median", fill = 'blue') +  
 scale\_y\_continuous(breaks = seq(0, 800000, by = 25000), labels = scales::comma) +  
 geom\_label(stat = "count", aes(label = ..count.., y = ..count..)) +  
 coord\_cartesian(ylim = c(0, 200000)) +  
 geom\_hline(yintercept = 163000, linetype = "dashed", color = "red")  
  
grid.arrange(year.sold, month.sold, widths=c(1,2))



##Factorize MSSUbClass  
str(df.combined$MSSubClass)

## int [1:2919] 60 20 60 70 60 50 20 60 50 190 ...

df.combined$MSSubClass <- as.factor(df.combined$MSSubClass)  
  
##MSSubClass Levels Vector  
MSSubClass <- c('20' = '1 Story 1946+', '30' = '1 Story 1945-', '40' = '1 Story Fin Attic All', '45' = '1.5 Story Unf All', '50' = '1.5 Story Fin All', '60' = '2 Story 1946+', '70' = '2 Story 1945-', '75' = '2.5 Story All', '80' = 'Split/Multi Level', '85' = 'Split Foyer', '90' = 'Duplex All Style/Age', '120' = '1 Story PUD 1946+', '150' = '1.5 Story PUD All', '160' = '2 Story PUD 1946+', '180' = 'PUD Multilevel', '190' = '2 Family Conversion')  
  
##Revalue for Ease  
df.combined$MSSubClass<- revalue(df.combined$MSSubClass, MSSubClass)

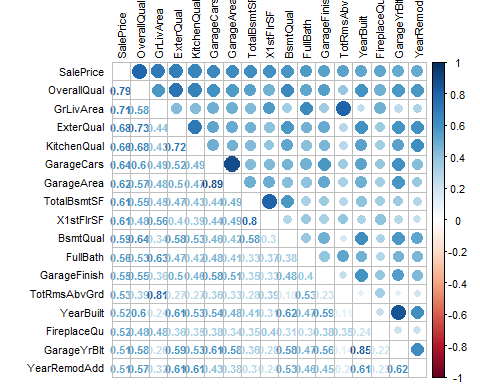
##Level Set after Data Clean & Imputation  
numeric.Vars <- which(sapply(df.combined, is.numeric))  
categoric.Vars <- which(sapply(df.combined, is.factor))  
  
cat(length(numeric.Vars), 'numeric variables &', length(categoric.Vars), 'categoric variables')

## 56 numeric variables & 23 categoric variables

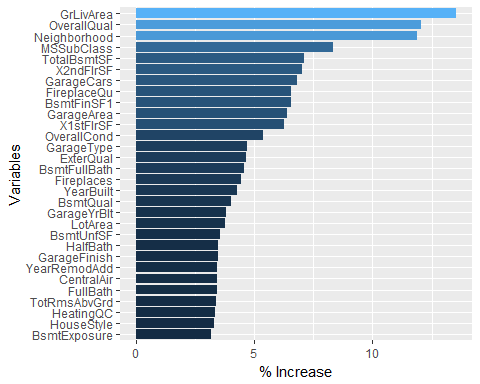
str(df.combined)

## 'data.frame': 2919 obs. of 79 variables:  
## $ MSSubClass : Factor w/ 16 levels "1 Story 1946+",..: 6 1 6 7 6 5 1 6 5 16 ...  
## $ MSZoning : Factor w/ 5 levels "C (all)","FV",..: 4 4 4 4 4 4 4 4 5 4 ...  
## $ LotFrontage : int 65 80 68 60 84 85 75 80 51 50 ...  
## $ LotArea : int 8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...  
## $ Street : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Alley : Factor w/ 3 levels "Grvl","None",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ LotShape : int 3 3 2 2 2 2 3 2 3 3 ...  
## $ LandContour : Factor w/ 4 levels "Bnk","HLS","Low",..: 4 4 4 4 4 4 4 4 4 4 ...  
## $ LotConfig : Factor w/ 5 levels "Corner","CulDSac",..: 5 3 5 1 3 5 5 1 5 1 ...  
## $ LandSlope : int 2 2 2 2 2 2 2 2 2 2 ...  
## $ Neighborhood : Factor w/ 25 levels "Blmngtn","Blueste",..: 6 25 6 7 14 12 21 17 18 4 ...  
## $ Condition1 : Factor w/ 9 levels "Artery","Feedr",..: 3 2 3 3 3 3 3 5 1 1 ...  
## $ Condition2 : Factor w/ 8 levels "Artery","Feedr",..: 3 3 3 3 3 3 3 3 3 1 ...  
## $ BldgType : Factor w/ 5 levels "1Fam","2fmCon",..: 1 1 1 1 1 1 1 1 1 2 ...  
## $ HouseStyle : Factor w/ 8 levels "1.5Fin","1.5Unf",..: 6 3 6 6 6 1 3 6 1 2 ...  
## $ OverallQual : int 7 6 7 7 8 5 8 7 7 5 ...  
## $ OverallCond : int 5 8 5 5 5 5 5 6 5 6 ...  
## $ YearBuilt : int 2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...  
## $ YearRemodAdd : int 2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...  
## $ RoofStyle : Factor w/ 6 levels "Flat","Gable",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ RoofMatl : Factor w/ 8 levels "ClyTile","CompShg",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ Exterior1st : Factor w/ 15 levels "AsbShng","AsphShn",..: 13 9 13 14 13 13 13 7 4 9 ...  
## $ Exterior2nd : Factor w/ 16 levels "AsbShng","AsphShn",..: 14 9 14 16 14 14 14 7 16 9 ...  
## $ MasVnrType : int 1 0 1 0 1 0 2 2 0 0 ...  
## $ MasVnrArea : num 196 0 162 0 350 0 186 240 0 0 ...  
## $ ExterQual : int 4 3 4 3 4 3 4 3 3 3 ...  
## $ ExterCond : int 3 3 3 3 3 3 3 3 3 3 ...  
## $ Foundation : Factor w/ 6 levels "BrkTil","CBlock",..: 3 2 3 1 3 6 3 2 1 1 ...  
## $ BsmtQual : int 4 4 4 3 4 4 5 4 3 3 ...  
## $ BsmtCond : int 3 3 3 4 3 3 3 3 3 3 ...  
## $ BsmtExposure : int 1 4 2 1 3 1 3 2 1 1 ...  
## $ BsmtFinType1 : int 6 5 6 5 6 6 6 5 1 6 ...  
## $ BsmtFinSF1 : num 706 978 486 216 655 ...  
## $ BsmtFinType2 : int 1 1 1 1 1 1 1 4 1 1 ...  
## $ BsmtFinSF2 : num 0 0 0 0 0 0 0 32 0 0 ...  
## $ BsmtUnfSF : num 150 284 434 540 490 64 317 216 952 140 ...  
## $ TotalBsmtSF : num 856 1262 920 756 1145 ...  
## $ Heating : Factor w/ 6 levels "Floor","GasA",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ HeatingQC : int 5 5 5 4 5 5 5 5 4 5 ...  
## $ CentralAir : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Electrical : Factor w/ 5 levels "FuseA","FuseF",..: 5 5 5 5 5 5 5 5 2 5 ...  
## $ X1stFlrSF : int 856 1262 920 961 1145 796 1694 1107 1022 1077 ...  
## $ X2ndFlrSF : int 854 0 866 756 1053 566 0 983 752 0 ...  
## $ LowQualFinSF : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ GrLivArea : int 1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...  
## $ BsmtFullBath : num 1 0 1 1 1 1 1 1 0 1 ...  
## $ BsmtHalfBath : num 0 1 0 0 0 0 0 0 0 0 ...  
## $ FullBath : int 2 2 2 1 2 1 2 2 2 1 ...  
## $ HalfBath : int 1 0 1 0 1 1 0 1 0 0 ...  
## $ BedroomAbvGr : int 3 3 3 3 4 1 3 3 2 2 ...  
## $ KitchenAbvGr : int 1 1 1 1 1 1 1 1 2 2 ...  
## $ KitchenQual : int 4 3 4 4 4 3 4 3 3 3 ...  
## $ TotRmsAbvGrd : int 8 6 6 7 9 5 7 7 8 5 ...  
## $ Functional : int 7 7 7 7 7 7 7 7 6 7 ...  
## $ Fireplaces : int 0 1 1 1 1 0 1 2 2 2 ...  
## $ FireplaceQu : int 0 3 3 4 3 0 4 3 3 3 ...  
## $ GarageType : Factor w/ 7 levels "2Types","Attchd",..: 2 2 2 6 2 2 2 2 6 2 ...  
## $ GarageYrBlt : int 2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 ...  
## $ GarageFinish : int 2 2 2 1 2 1 2 2 1 2 ...  
## $ GarageCars : num 2 2 2 3 3 2 2 2 2 1 ...  
## $ GarageArea : num 548 460 608 642 836 480 636 484 468 205 ...  
## $ GarageQual : int 3 3 3 3 3 3 3 3 2 4 ...  
## $ GarageCond : int 3 3 3 3 3 3 3 3 3 3 ...  
## $ PavedDrive : int 2 2 2 2 2 2 2 2 2 2 ...  
## $ WoodDeckSF : int 0 298 0 0 192 40 255 235 90 0 ...  
## $ OpenPorchSF : int 61 0 42 35 84 30 57 204 0 4 ...  
## $ EnclosedPorch: int 0 0 0 272 0 0 0 228 205 0 ...  
## $ X3SsnPorch : int 0 0 0 0 0 320 0 0 0 0 ...  
## $ ScreenPorch : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ PoolArea : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ PoolQC : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Fence : Factor w/ 5 levels "GdPrv","GdWo",..: 5 5 5 5 5 3 5 5 5 5 ...  
## $ MiscFeature : Factor w/ 5 levels "Gar2","None",..: 2 2 2 2 2 4 2 4 2 2 ...  
## $ MiscVal : int 0 0 0 0 0 700 0 350 0 0 ...  
## $ MoSold : Factor w/ 12 levels "1","2","3","4",..: 2 5 9 2 12 10 8 11 4 1 ...  
## $ YrSold : int 2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...  
## $ SaleType : Factor w/ 9 levels "COD","Con","ConLD",..: 9 9 9 9 9 9 9 9 9 9 ...  
## $ SaleCondition: Factor w/ 6 levels "Abnorml","AdjLand",..: 5 5 5 1 5 5 5 5 1 5 ...  
## $ SalePrice : int 208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...

##Level Set High Correlations  
df.numeric.Vars <- df.combined[, numeric.Vars]  
correlation.numeric.Vars <- cor(df.numeric.Vars, use = "pairwise.complete.obs")  
correlation.sorted <- as.matrix(sort(correlation.numeric.Vars[, 'SalePrice'], decreasing = TRUE))  
  
correlation.high <- names(which(apply(correlation.sorted, 1, function(x) abs(x) > 0.5)))  
correlation.numeric.Vars <- correlation.numeric.Vars[correlation.high, correlation.high]  
  
corrplot.mixed(correlation.numeric.Vars, tl.col = "black", tl.pos = "lt", tl.cex = 0.7, cl.cex = 0.7, number.cex = 0.7)



##Quick RandomForest to Identify Most Important Variables(Numeric & Categorical)  
set.seed(2018)  
RandomForest.draft <- randomForest(x = df.combined[1:1460, -79], y = df.combined$SalePrice[1:1460], ntree = 100, importance = TRUE)  
RandomForest.important <- importance(RandomForest.draft)  
df.important <- data.frame(Variables = row.names(RandomForest.important), MSE = RandomForest.important[, 1])  
df.important <- df.important[order(df.important$MSE, decreasing = TRUE),]  
  
##Visualize Quick RandomForest (df.important)  
ggplot(df.important[1:30,], aes(x = reorder(Variables, MSE), y = MSE, fill = MSE)) +  
 geom\_bar(stat = 'identity') +  
 labs(x = 'Variables', y = '% Increase') +  
 coord\_flip() +  
 theme(legend.position = "none")

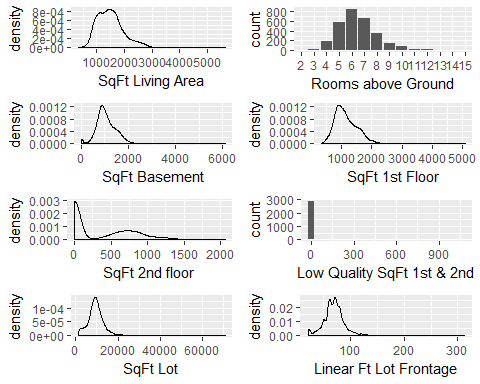


##EDA of GrLivArea & Surface Related Variables  
s1 <- ggplot(data = df.combined, aes(x = GrLivArea)) +  
 geom\_density() + labs(x = 'SqFt Living Area')  
s2 <- ggplot(data = df.combined, aes(x = as.factor(TotRmsAbvGrd))) +  
 geom\_histogram(stat = 'count') + labs(x = 'Rooms above Ground')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

s3 <- ggplot(data = df.combined, aes(x = X1stFlrSF)) +  
 geom\_density() + labs(x = 'SqFt 1st Floor')  
s4 <- ggplot(data = df.combined, aes(x = X2ndFlrSF)) +  
 geom\_density() + labs(x='SqFt 2nd floor')  
s5 <- ggplot(data = df.combined, aes(x = TotalBsmtSF)) +  
 geom\_density() + labs(x = 'SqFt Basement')  
s6 <- ggplot(data = df.combined[df.combined$LotArea < 100000,], aes(x=LotArea)) +  
 geom\_density() + labs(x = 'SqFt Lot')  
s7 <- ggplot(data = df.combined, aes(x=LotFrontage)) +  
 geom\_density() + labs(x = 'Linear Ft Lot Frontage')  
s8 <- ggplot(data = df.combined, aes(x = LowQualFinSF)) +  
 geom\_histogram() + labs(x = 'Low Quality SqFt 1st & 2nd')  
  
d.layout <- matrix(c(1, 2, 5, 3, 4, 8, 6, 7), 4, 2, byrow=TRUE)  
multiplot(s1, s2, s3, s4, s5, s6, s7, s8, layout = d.layout)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



##Investigating Correlation between GrLivArea to X1stFlrSF, X2ndFlrSF & LowQualFinSF  
cor(df.combined$GrLivArea, (df.combined$X1stFlrSF + df.combined$X2ndFlrSF + df.combined$LowQualFinSF))

## [1] 1

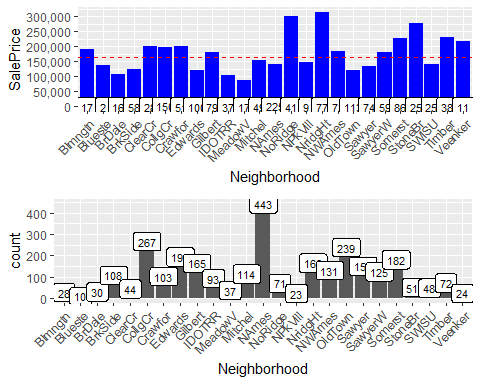
head(df.combined[df.combined$LowQualFinSF > 0, c('GrLivArea', 'X1stFlrSF', 'X2ndFlrSF', 'LowQualFinSF')])

## GrLivArea X1stFlrSF X2ndFlrSF LowQualFinSF  
## 52 1176 816 0 360  
## 89 1526 1013 0 513  
## 126 754 520 0 234  
## 171 1382 854 0 528  
## 186 3608 1518 1518 572  
## 188 1656 808 704 144

##EDA of Neighborhood Variable  
n1 <-ggplot(df.combined[!is.na(df.combined$SalePrice), ], aes(x = Neighborhood, y = SalePrice)) +  
 geom\_bar(stat = 'summary', fun.y = "median", fill = 'blue') +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_y\_continuous(breaks = seq(0, 800000, by = 50000), labels = scales::comma) +  
 geom\_label(stat = "count", aes(label = ..count.., y = ..count..), size = 3) +  
 geom\_hline(yintercept = 163000, linetype = "dashed", color = "red")  
  
n2 <- ggplot(data = df.combined, aes(x = Neighborhood)) +  
 geom\_histogram(stat = 'count') +  
 geom\_label(stat = "count", aes(label = ..count.., y = ..count..), size = 3) +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))

## Warning: Ignoring unknown parameters: binwidth, bins, pad

grid.arrange(n1, n2)



##EDA of OverallQuall & Quality Variables  
q1 <- ggplot(data = df.combined, aes(x = as.factor(OverallQual))) +  
 geom\_histogram(stat = 'count')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

q2 <- ggplot(data = df.combined, aes(x=as.factor(ExterQual))) +  
 geom\_histogram(stat='count')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

q3 <- ggplot(data = df.combined, aes(x=as.factor(BsmtQual))) +  
 geom\_histogram(stat='count')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

q4 <- ggplot(data = df.combined, aes(x=as.factor(KitchenQual))) +  
 geom\_histogram(stat='count')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

q5 <- ggplot(data = df.combined, aes(x=as.factor(GarageQual))) +  
 geom\_histogram(stat='count')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

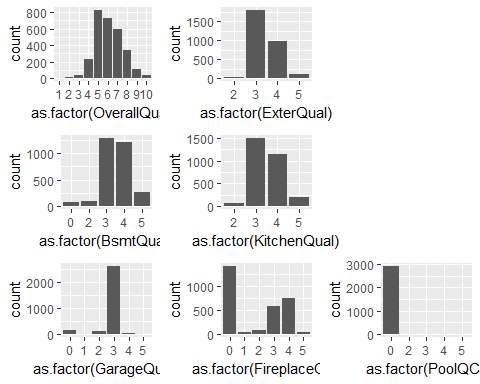
q6 <- ggplot(data = df.combined, aes(x=as.factor(FireplaceQu))) +  
 geom\_histogram(stat='count')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

q7 <- ggplot(data = df.combined, aes(x=as.factor(PoolQC))) +  
 geom\_histogram(stat='count')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

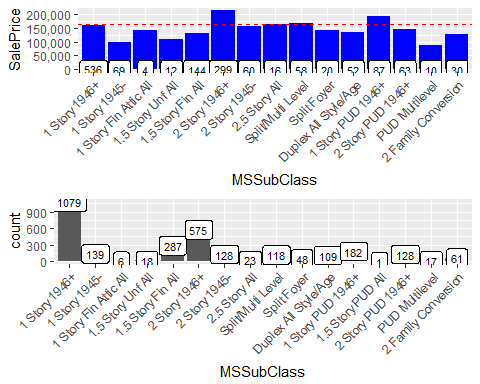
layout <- matrix(c(1,2,8,3,4,8,5,6,7),3,3,byrow=TRUE)  
multiplot(q1, q2, q3, q4, q5, q6, q7, layout=layout)



##EDA of MSSubCLass  
ms1 <- ggplot(df.combined[!is.na(df.combined$SalePrice),], aes(x = MSSubClass, y = SalePrice)) +  
 geom\_bar(stat = 'summary', fun.y = "median", fill = 'blue') +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_y\_continuous(breaks = seq(0, 800000, by=50000), labels = comma) +  
 geom\_label(stat = "count", aes(label = ..count.., y = ..count..), size = 3) +  
 geom\_hline(yintercept = 163000, linetype = "dashed", color = "red")   
  
ms2 <- ggplot(data = df.combined, aes(x = MSSubClass)) +  
 geom\_histogram(stat = 'count') +  
 geom\_label(stat = "count", aes(label = ..count.., y = ..count..), size = 3) +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))

## Warning: Ignoring unknown parameters: binwidth, bins, pad

grid.arrange(ms1, ms2)



##EDA of Garage Variables  
g1 <- ggplot(data = df.combined[df.combined$GarageCars !=0,], aes(x = GarageYrBlt)) +   
 geom\_histogram() +   
 xlim(1890, 2015)  
g2 <- ggplot(data = df.combined, aes(x = as.factor(GarageCars))) +  
 geom\_histogram(stat = 'count')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

g3 <- ggplot(data = df.combined, aes(x = GarageArea)) +  
 geom\_density()  
g4 <- ggplot(data = df.combined, aes(x = as.factor(GarageCond))) +  
 geom\_histogram(stat = 'count')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

g5 <- ggplot(data = df.combined, aes(x = GarageType)) +  
 geom\_histogram(stat = 'count')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

g6 <- ggplot(data = df.combined, aes(x = as.factor(GarageQual))) +  
 geom\_histogram(stat = 'count')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

g7 <- ggplot(data = df.combined, aes(x = as.factor(GarageFinish))) +  
 geom\_histogram(stat = 'count')

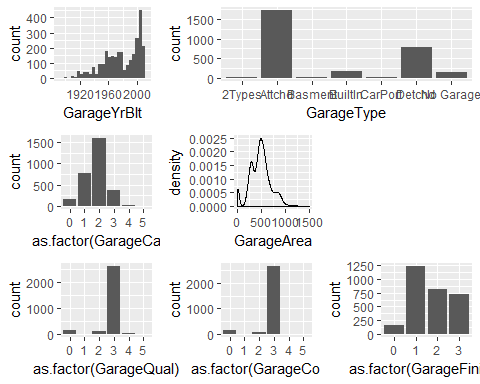
## Warning: Ignoring unknown parameters: binwidth, bins, pad

layout <- matrix(c(1, 5, 5, 2, 3, 8, 6, 4, 7), 3 , 3, byrow = TRUE)  
multiplot(g1, g2, g3, g4, g5, g6, g7, layout = layout)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 1 rows containing non-finite values (stat\_bin).

## Warning: Removed 1 rows containing missing values (geom\_bar).



##EDA of Basement Variables  
b1 <- ggplot(data = df.combined, aes(BsmtFinSF1)) +   
 geom\_histogram() +  
 labs(x = 'Type 1 Finished SqFt')  
b2 <- ggplot(data = df.combined, aes(x = BsmtFinSF2)) +  
 geom\_histogram() +  
 labs(x = 'Type 2 Finished SqFt')  
b3 <- ggplot(data = df.combined, aes(x = BsmtUnfSF)) +   
 geom\_histogram() +   
 labs(x = 'Unfinished SqFt')  
b4 <- ggplot(data = df.combined, aes(x = as.factor(BsmtFinType1))) +  
 geom\_histogram(stat = 'count') +  
 labs(x = 'Type 1 Finished Area Rating')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

b5 <- ggplot(data = df.combined, aes(x = as.factor(BsmtFinType2))) +  
 geom\_histogram(stat = 'count') +  
 labs(x = 'Type 2 Finished Area Rating')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

b6 <- ggplot(data = df.combined, aes(x = as.factor(BsmtQual))) +  
 geom\_histogram(stat = 'count') +  
 labs(x = 'Height of Bsmt')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

b7 <- ggplot(data = df.combined, aes(x = as.factor(BsmtCond))) +  
 geom\_histogram(stat = 'count') +  
 labs(x = 'Bsmt Condition Rating')

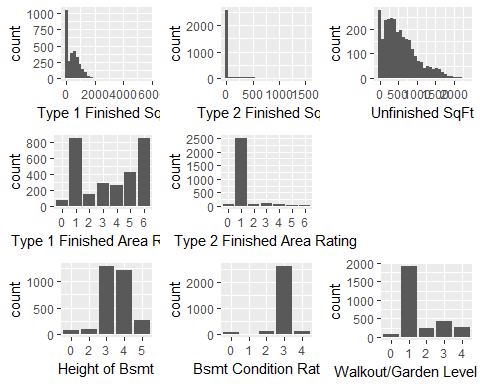
## Warning: Ignoring unknown parameters: binwidth, bins, pad

b8 <- ggplot(data = df.combined, aes(x = as.factor(BsmtExposure))) +  
 geom\_histogram(stat = 'count') +  
 labs(x = 'Walkout/Garden Level Walls')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

layout <- matrix(c(1, 2, 3, 4, 5, 9, 6, 7, 8), 3, 3, byrow = TRUE)  
multiplot(b1, b2, b3, b4, b5, b6, b7, b8, layout = layout)

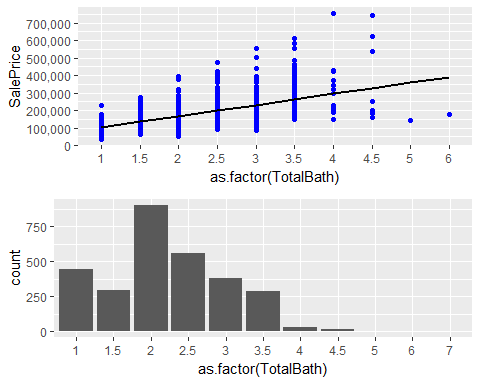
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



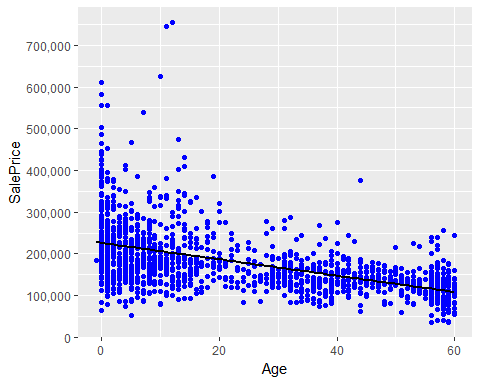
##Feature Engineering  
  
##Total Bathrooms  
df.combined$TotalBath <- df.combined$FullBath + (df.combined$HalfBath\*0.5) + df.combined$BsmtFullBath + (df.combined$BsmtHalfBath\*0.5)  
  
tb1 <- ggplot(data = df.combined[!is.na(df.combined$SalePrice), ], aes(x = as.factor(TotalBath), y = SalePrice)) +  
 geom\_point(col = 'blue') + geom\_smooth(method = "lm", se = FALSE, color = "black", aes(group = 1)) +  
 scale\_y\_continuous(breaks = seq(0, 800000, by = 100000), labels = comma)  
  
tb2 <- ggplot(data = df.combined, aes(x = as.factor(TotalBath))) +  
 geom\_histogram(stat = 'count')

## Warning: Ignoring unknown parameters: binwidth, bins, pad

grid.arrange(tb1, tb2)



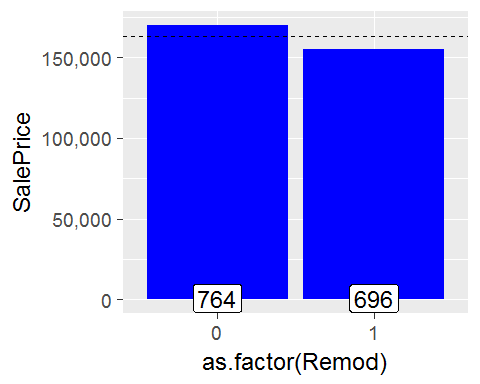
##Age, Remodel & New Variables  
df.combined$Remod <- ifelse(df.combined$YearBuilt == df.combined$YearRemodAdd, 0, 1) ## 0 = NA Remodel, 1 = Remodel  
df.combined$Age <- as.numeric(df.combined$YrSold) - df.combined$YearRemodAdd  
  
ggplot(data = df.combined[!is.na(df.combined$SalePrice), ], aes(x = Age, y = SalePrice)) +  
 geom\_point(col = 'blue') +  
 geom\_smooth(method = "lm", se = FALSE, color = "black", aes(group = 1)) +  
 scale\_y\_continuous(breaks = seq(0, 800000, by = 100000), labels = comma)



cor(df.combined$SalePrice[!is.na(df.combined$SalePrice)], df.combined$Age[!is.na(df.combined$SalePrice)])

## [1] -0.5090787

ggplot(df.combined[!is.na(df.combined$SalePrice),], aes(x = as.factor(Remod), y = SalePrice)) +  
 geom\_bar(stat = 'summary', fun.y = "median", fill = 'blue') +  
 geom\_label(stat = "count", aes(label = ..count.., y = ..count..), size = 6) +  
 scale\_y\_continuous(breaks = seq(0, 800000, by = 50000), labels = comma) +  
 theme\_grey(base\_size = 18) +  
 geom\_hline(yintercept = 163000, linetype = "dashed")

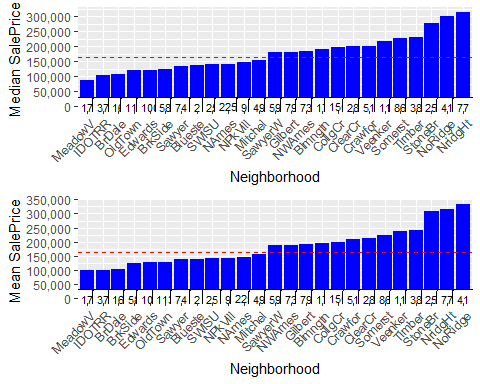


df.combined$New <- ifelse(df.combined$YrSold == df.combined$YearBuilt, 1, 0) ##1 = New, 0 = Not New  
table(df.combined$New)

##   
## 0 1   
## 2803 116

##Factorize YrSold  
df.combined$YrSold <- as.factor(df.combined$YrSold)

##Binning Neighborhoods  
nb1 <- ggplot(df.combined[!is.na(df.combined$SalePrice), ], aes(x = reorder(Neighborhood, SalePrice, FUN = median), y = SalePrice)) +  
 geom\_bar(stat = 'summary', fun.y = "median", fill = 'blue') +   
 labs(x = 'Neighborhood', y = 'Median SalePrice') +  
 theme(axis.text.x = element\_text(angle = 45, hjust =1)) +  
 scale\_y\_continuous(breaks = seq(0, 800000, by = 50000), labels = comma) +   
 geom\_label(stat = "count", aes(label = ..count.., y = ..count..), size = 3) +  
 geom\_hline(yintercept = 163000, linetype = "dashed", color = "red")  
  
nb2 <- ggplot(df.combined[!is.na(df.combined$SalePrice), ], aes(x = reorder(Neighborhood, SalePrice, FUN = mean), y = SalePrice)) +  
 geom\_bar(stat = 'summary', fun.y = "mean", fill = 'blue') +   
 labs(x = 'Neighborhood', y = 'Mean SalePrice') +  
 theme(axis.text.x = element\_text(angle = 45, hjust =1)) +  
 scale\_y\_continuous(breaks = seq(0, 800000, by = 50000), labels = comma) +   
 geom\_label(stat = "count", aes(label = ..count.., y = ..count..), size = 3) +  
 geom\_hline(yintercept = 163000, linetype = "dashed", color = "red")  
  
grid.arrange(nb1, nb2)



##Categorizing Neighborhood Wealth (Poor - 0, Middle - 1, Wealthy - 2)  
df.combined$NeighborhoodWealth[df.combined$Neighborhood %in% c('StoneBr', 'NridgHt', 'NoRidge')] <- 2  
df.combined$NeighborhoodWealth[!df.combined$Neighborhood %in% c('MeadowV', 'IDOTRR', 'BrDale', 'StoneBr', 'NridgHt', 'NoRidge')] <- 1  
df.combined$NeighborhoodWealth[df.combined$Neighborhood %in% c('MeadowV', 'IDOTRR', 'BrDale')] <- 0  
table(df.combined$NeighborhoodWealth)

##   
## 0 1 2   
## 160 2471 288

sum(table(df.combined$NeighborhoodWealth))

## [1] 2919

## Total SqFt  
df.combined$TotalSqFt <- df.combined$GrLivArea + df.combined$TotalBsmtSF  
  
cor(df.combined$SalePrice, df.combined$TotalSqFt, use = "pairwise.complete.obs")

## [1] 0.7789588

##Porch Variables  
df.combined$TotalPorchSqFt <- df.combined$OpenPorchSF + df.combined$EnclosedPorch + df.combined$X3SsnPorch + df.combined$ScreenPorch  
  
cor(df.combined$SalePrice, df.combined$TotalPorchSqFt, use = "pairwise.complete.obs")

## [1] 0.1957389

##Data Prep for Modeling  
  
##Remove Highly Correlated Variables  
rmVars <- c('YearRemodAdd', 'GarageYrBlt', 'GarageArea', 'GarageCond', 'TotalBsmtSF', 'TotRmsAbvGrd')  
  
df.combined <- df.combined[, !(names(df.combined) %in% rmVars)]  
  
##Remove Outliers  
df.combined <- df.combined[-c(524, 1299),]

##Prepping Predictor Variables  
  
numeric.VarNames <- numeric.VarNames[!(numeric.VarNames %in% c('MSSubClass', 'MoSold', 'YrSold', 'SalePrice', 'OverallQual', 'OverallCond'))]  
numeric.VarNames <- append(numeric.VarNames, c('Age', 'TotalPorchSqFt', 'TotalBath', 'TotalSqFt'))  
  
df.numeric <- df.combined[, names(df.combined) %in% numeric.VarNames]  
str(df.numeric)

## 'data.frame': 2917 obs. of 30 variables:  
## $ LotFrontage : int 65 80 68 60 84 85 75 80 51 50 ...  
## $ LotArea : int 8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...  
## $ YearBuilt : int 2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...  
## $ MasVnrArea : num 196 0 162 0 350 0 186 240 0 0 ...  
## $ BsmtFinSF1 : num 706 978 486 216 655 ...  
## $ BsmtFinSF2 : num 0 0 0 0 0 0 0 32 0 0 ...  
## $ BsmtUnfSF : num 150 284 434 540 490 64 317 216 952 140 ...  
## $ X1stFlrSF : int 856 1262 920 961 1145 796 1694 1107 1022 1077 ...  
## $ X2ndFlrSF : int 854 0 866 756 1053 566 0 983 752 0 ...  
## $ LowQualFinSF : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ GrLivArea : int 1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...  
## $ BsmtFullBath : num 1 0 1 1 1 1 1 1 0 1 ...  
## $ BsmtHalfBath : num 0 1 0 0 0 0 0 0 0 0 ...  
## $ FullBath : int 2 2 2 1 2 1 2 2 2 1 ...  
## $ HalfBath : int 1 0 1 0 1 1 0 1 0 0 ...  
## $ BedroomAbvGr : int 3 3 3 3 4 1 3 3 2 2 ...  
## $ KitchenAbvGr : int 1 1 1 1 1 1 1 1 2 2 ...  
## $ Fireplaces : int 0 1 1 1 1 0 1 2 2 2 ...  
## $ GarageCars : num 2 2 2 3 3 2 2 2 2 1 ...  
## $ WoodDeckSF : int 0 298 0 0 192 40 255 235 90 0 ...  
## $ OpenPorchSF : int 61 0 42 35 84 30 57 204 0 4 ...  
## $ EnclosedPorch : int 0 0 0 272 0 0 0 228 205 0 ...  
## $ X3SsnPorch : int 0 0 0 0 0 320 0 0 0 0 ...  
## $ ScreenPorch : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ PoolArea : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ MiscVal : int 0 0 0 0 0 700 0 350 0 0 ...  
## $ TotalBath : num 3.5 2.5 3.5 2 3.5 2.5 3 3.5 2 2 ...  
## $ Age : num 5 31 6 36 8 14 2 36 58 58 ...  
## $ TotalSqFt : num 2566 2524 2706 2473 3343 ...  
## $ TotalPorchSqFt: int 61 0 42 307 84 350 57 432 205 4 ...

df.categoric <- df.combined[, !(names(df.combined) %in% numeric.VarNames)]  
df.categoric <- df.categoric[, names(df.categoric) != 'SalePrice']  
str(df.categoric)

## 'data.frame': 2917 obs. of 49 variables:  
## $ MSSubClass : Factor w/ 16 levels "1 Story 1946+",..: 6 1 6 7 6 5 1 6 5 16 ...  
## $ MSZoning : Factor w/ 5 levels "C (all)","FV",..: 4 4 4 4 4 4 4 4 5 4 ...  
## $ Street : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Alley : Factor w/ 3 levels "Grvl","None",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ LotShape : int 3 3 2 2 2 2 3 2 3 3 ...  
## $ LandContour : Factor w/ 4 levels "Bnk","HLS","Low",..: 4 4 4 4 4 4 4 4 4 4 ...  
## $ LotConfig : Factor w/ 5 levels "Corner","CulDSac",..: 5 3 5 1 3 5 5 1 5 1 ...  
## $ LandSlope : int 2 2 2 2 2 2 2 2 2 2 ...  
## $ Neighborhood : Factor w/ 25 levels "Blmngtn","Blueste",..: 6 25 6 7 14 12 21 17 18 4 ...  
## $ Condition1 : Factor w/ 9 levels "Artery","Feedr",..: 3 2 3 3 3 3 3 5 1 1 ...  
## $ Condition2 : Factor w/ 8 levels "Artery","Feedr",..: 3 3 3 3 3 3 3 3 3 1 ...  
## $ BldgType : Factor w/ 5 levels "1Fam","2fmCon",..: 1 1 1 1 1 1 1 1 1 2 ...  
## $ HouseStyle : Factor w/ 8 levels "1.5Fin","1.5Unf",..: 6 3 6 6 6 1 3 6 1 2 ...  
## $ OverallQual : int 7 6 7 7 8 5 8 7 7 5 ...  
## $ OverallCond : int 5 8 5 5 5 5 5 6 5 6 ...  
## $ RoofStyle : Factor w/ 6 levels "Flat","Gable",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ RoofMatl : Factor w/ 8 levels "ClyTile","CompShg",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ Exterior1st : Factor w/ 15 levels "AsbShng","AsphShn",..: 13 9 13 14 13 13 13 7 4 9 ...  
## $ Exterior2nd : Factor w/ 16 levels "AsbShng","AsphShn",..: 14 9 14 16 14 14 14 7 16 9 ...  
## $ MasVnrType : int 1 0 1 0 1 0 2 2 0 0 ...  
## $ ExterQual : int 4 3 4 3 4 3 4 3 3 3 ...  
## $ ExterCond : int 3 3 3 3 3 3 3 3 3 3 ...  
## $ Foundation : Factor w/ 6 levels "BrkTil","CBlock",..: 3 2 3 1 3 6 3 2 1 1 ...  
## $ BsmtQual : int 4 4 4 3 4 4 5 4 3 3 ...  
## $ BsmtCond : int 3 3 3 4 3 3 3 3 3 3 ...  
## $ BsmtExposure : int 1 4 2 1 3 1 3 2 1 1 ...  
## $ BsmtFinType1 : int 6 5 6 5 6 6 6 5 1 6 ...  
## $ BsmtFinType2 : int 1 1 1 1 1 1 1 4 1 1 ...  
## $ Heating : Factor w/ 6 levels "Floor","GasA",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ HeatingQC : int 5 5 5 4 5 5 5 5 4 5 ...  
## $ CentralAir : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Electrical : Factor w/ 5 levels "FuseA","FuseF",..: 5 5 5 5 5 5 5 5 2 5 ...  
## $ KitchenQual : int 4 3 4 4 4 3 4 3 3 3 ...  
## $ Functional : int 7 7 7 7 7 7 7 7 6 7 ...  
## $ FireplaceQu : int 0 3 3 4 3 0 4 3 3 3 ...  
## $ GarageType : Factor w/ 7 levels "2Types","Attchd",..: 2 2 2 6 2 2 2 2 6 2 ...  
## $ GarageFinish : int 2 2 2 1 2 1 2 2 1 2 ...  
## $ GarageQual : int 3 3 3 3 3 3 3 3 2 4 ...  
## $ PavedDrive : int 2 2 2 2 2 2 2 2 2 2 ...  
## $ PoolQC : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Fence : Factor w/ 5 levels "GdPrv","GdWo",..: 5 5 5 5 5 3 5 5 5 5 ...  
## $ MiscFeature : Factor w/ 5 levels "Gar2","None",..: 2 2 2 2 2 4 2 4 2 2 ...  
## $ MoSold : Factor w/ 12 levels "1","2","3","4",..: 2 5 9 2 12 10 8 11 4 1 ...  
## $ YrSold : Factor w/ 5 levels "2006","2007",..: 3 2 3 1 3 4 2 4 3 3 ...  
## $ SaleType : Factor w/ 9 levels "COD","Con","ConLD",..: 9 9 9 9 9 9 9 9 9 9 ...  
## $ SaleCondition : Factor w/ 6 levels "Abnorml","AdjLand",..: 5 5 5 1 5 5 5 5 1 5 ...  
## $ Remod : num 0 0 1 1 0 1 1 0 1 1 ...  
## $ New : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodWealth: num 1 1 1 1 2 1 1 1 1 1 ...

cat(length(df.numeric), 'numeric variables &', length(df.categoric), 'categoric variables')

## 30 numeric variables & 49 categoric variables

##Skewness of Numeric Variables  
for(i in 1:ncol(df.numeric)){  
 if(abs(skew(df.numeric[,i])) > 0.8){  
 df.numeric[,i] <- log(df.numeric[,i] + 1)  
 }  
}  
  
##Normalization of Numeric Variables  
Predictor.Vars <- preProcess(df.numeric, method = c("center", "scale"))  
print(Predictor.Vars)

## Created from 2917 samples and 30 variables  
##   
## Pre-processing:  
## - centered (30)  
## - ignored (0)  
## - scaled (30)

df.normal <- predict(Predictor.Vars, df.numeric)  
dim(df.normal)

## [1] 2917 30

##Encoding of Categoric Varibles  
df.dummy <- as.data.frame(model.matrix(~.-1, df.categoric))  
dim(df.dummy)

## [1] 2917 201

##Removing Levels with Few/None Values  
  
##Absent Values in Test Set  
Values.Absent.Test <- which(colSums(df.dummy[1459:2917, ]) == 0)  
colnames(df.dummy[Values.Absent.Test])

## [1] "Condition2RRAe" "Condition2RRAn" "Condition2RRNn"   
## [4] "HouseStyle2.5Fin" "RoofMatlMembran" "RoofMatlMetal"   
## [7] "RoofMatlRoll" "Exterior1stImStucc" "Exterior1stStone"   
## [10] "Exterior2ndOther" "HeatingOthW" "ElectricalMix"   
## [13] "MiscFeatureTenC"

##Removing Predictor Values  
df.dummy <- df.dummy[, -Values.Absent.Test]  
  
##Absent Values in Train Set  
Values.Absent.Train <- which(colSums(df.dummy[1:1458, ]) == 0)  
colnames(df.dummy[Values.Absent.Train])

## [1] "MSSubClass1.5 Story PUD All"

##Removing Predictor Values  
df.dummy <- df.dummy[, -Values.Absent.Train]  
  
##Few Values (<10) in Train Set  
Values.Few.Train <- which(colSums(df.dummy[1:1458, ]) <10)  
colnames(df.dummy[Values.Few.Train])

## [1] "MSSubClass1 Story Fin Attic All" "LotConfigFR3"   
## [3] "NeighborhoodBlueste" "NeighborhoodNPkVill"   
## [5] "Condition1PosA" "Condition1RRNe"   
## [7] "Condition1RRNn" "Condition2Feedr"   
## [9] "Condition2PosA" "Condition2PosN"   
## [11] "RoofStyleMansard" "RoofStyleShed"   
## [13] "RoofMatlWdShake" "RoofMatlWdShngl"   
## [15] "Exterior1stAsphShn" "Exterior1stBrkComm"   
## [17] "Exterior1stCBlock" "Exterior2ndAsphShn"   
## [19] "Exterior2ndBrk Cmn" "Exterior2ndCBlock"   
## [21] "Exterior2ndStone" "FoundationStone"   
## [23] "FoundationWood" "HeatingGrav"   
## [25] "HeatingWall" "ElectricalFuseP"   
## [27] "GarageTypeCarPort" "MiscFeatureOthr"   
## [29] "SaleTypeCon" "SaleTypeConLD"   
## [31] "SaleTypeConLI" "SaleTypeConLw"   
## [33] "SaleTypeCWD" "SaleTypeOth"   
## [35] "SaleConditionAdjLand"

##Removing Predictor Values  
df.dummy <- df.dummy[, -Values.Few.Train]  
dim(df.dummy)

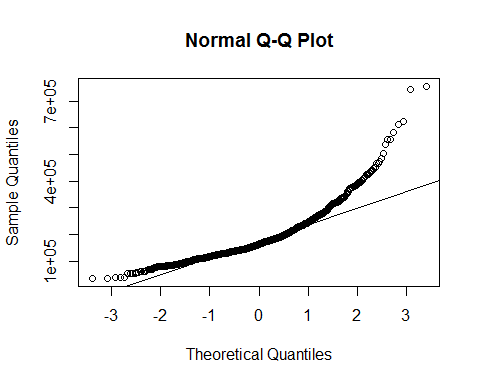
## [1] 2917 152

##Combine Predictors into Data Frame   
df.combined.predictors <- cbind(df.normal, df.dummy)

##Verifying Skewness of Response Variable (SalePrice)  
skew(df.combined$SalePrice)

## [1] 1.877427

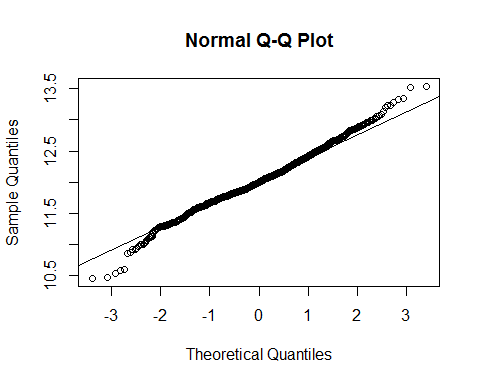
qqnorm(df.combined$SalePrice)  
qqline(df.combined$SalePrice)



##Normally Distribute SalePrice  
df.combined$SalePrice <- log(df.combined$SalePrice)  
  
##Verifying Skewness of Normalized Response Variable (SalePrice)  
skew(df.combined$SalePrice)

## [1] 0.1213182

qqnorm(df.combined$SalePrice)  
qqline(df.combined$SalePrice)



##Saving Train & Test Sets  
train.set <- df.combined.predictors[!is.na(df.combined$SalePrice), ]  
str(train.set)

## 'data.frame': 1458 obs. of 182 variables:  
## $ LotFrontage : num -0.0497 0.5752 0.0859 -0.2901 0.7223 ...  
## $ LotArea : num -0.102 0.149 0.462 0.139 0.928 ...  
## $ YearBuilt : num 1.047 0.156 0.981 -1.858 0.948 ...  
## $ MasVnrArea : num 1.225 -0.793 1.152 -0.793 1.445 ...  
## $ BsmtFinSF1 : num 0.785 0.894 0.66 0.388 0.759 ...  
## $ BsmtFinSF2 : num -0.363 -0.363 -0.363 -0.363 -0.363 ...  
## $ BsmtUnfSF : num -0.3248 0.0149 0.241 0.3577 0.3058 ...  
## $ X1stFlrSF : num -0.778 0.43 -0.554 -0.418 0.127 ...  
## $ X2ndFlrSF : num 1.198 -0.863 1.202 1.161 1.262 ...  
## $ LowQualFinSF : num -0.117 -0.117 -0.117 -0.117 -0.117 ...  
## $ GrLivArea : num 0.57 -0.369 0.705 0.583 1.347 ...  
## $ BsmtFullBath : num 1.09 -0.819 1.09 1.09 1.09 ...  
## $ BsmtHalfBath : num -0.252 3.887 -0.252 -0.252 -0.252 ...  
## $ FullBath : num 0.783 0.783 0.783 -1.027 0.783 ...  
## $ HalfBath : num 1.233 -0.755 1.233 -0.755 1.233 ...  
## $ BedroomAbvGr : num 0.17 0.17 0.17 0.17 1.39 ...  
## $ KitchenAbvGr : num -0.201 -0.201 -0.201 -0.201 -0.201 ...  
## $ Fireplaces : num -0.925 0.626 0.626 0.626 0.626 ...  
## $ GarageCars : num 0.308 0.308 0.308 1.62 1.62 ...  
## $ WoodDeckSF : num -0.944 1.255 -0.944 -0.944 1.086 ...  
## $ OpenPorchSF : num 0.834 -1.082 0.664 0.582 0.981 ...  
## $ EnclosedPorch : num -0.428 -0.428 -0.428 2.737 -0.428 ...  
## $ X3SsnPorch : num -0.113 -0.113 -0.113 -0.113 -0.113 ...  
## $ ScreenPorch : num -0.309 -0.309 -0.309 -0.309 -0.309 ...  
## $ PoolArea : num -0.0641 -0.0641 -0.0641 -0.0641 -0.0641 ...  
## $ MiscVal : num -0.189 -0.189 -0.189 -0.189 -0.189 ...  
## $ TotalBath : num 1.59 0.351 1.59 -0.269 1.59 ...  
## $ Age : num -0.888 0.357 -0.84 0.596 -0.744 ...  
## $ TotalSqFt : num 0.174 0.121 0.345 0.055 1.025 ...  
## $ TotalPorchSqFt : num 0.434 -1.453 0.267 1.167 0.579 ...  
## $ MSSubClass1 Story 1946+ : num 0 1 0 0 0 0 1 0 0 0 ...  
## $ MSSubClass1 Story 1945- : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClass1.5 Story Unf All : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClass1.5 Story Fin All : num 0 0 0 0 0 1 0 0 1 0 ...  
## $ MSSubClass2 Story 1946+ : num 1 0 1 0 1 0 0 1 0 0 ...  
## $ MSSubClass2 Story 1945- : num 0 0 0 1 0 0 0 0 0 0 ...  
## $ MSSubClass2.5 Story All : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClassSplit/Multi Level : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClassSplit Foyer : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClassDuplex All Style/Age: num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClass1 Story PUD 1946+ : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClass2 Story PUD 1946+ : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClassPUD Multilevel : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClass2 Family Conversion : num 0 0 0 0 0 0 0 0 0 1 ...  
## $ MSZoningFV : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSZoningRH : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSZoningRL : num 1 1 1 1 1 1 1 1 0 1 ...  
## $ MSZoningRM : num 0 0 0 0 0 0 0 0 1 0 ...  
## $ Street : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ AlleyNone : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ AlleyPave : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ LotShape : num 3 3 2 2 2 2 3 2 3 3 ...  
## $ LandContourHLS : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ LandContourLow : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ LandContourLvl : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ LotConfigCulDSac : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ LotConfigFR2 : num 0 1 0 0 1 0 0 0 0 0 ...  
## $ LotConfigInside : num 1 0 1 0 0 1 1 0 1 0 ...  
## $ LandSlope : num 2 2 2 2 2 2 2 2 2 2 ...  
## $ NeighborhoodBrDale : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodBrkSide : num 0 0 0 0 0 0 0 0 0 1 ...  
## $ NeighborhoodClearCr : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodCollgCr : num 1 0 1 0 0 0 0 0 0 0 ...  
## $ NeighborhoodCrawfor : num 0 0 0 1 0 0 0 0 0 0 ...  
## $ NeighborhoodEdwards : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodGilbert : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodIDOTRR : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodMeadowV : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodMitchel : num 0 0 0 0 0 1 0 0 0 0 ...  
## $ NeighborhoodNAmes : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodNoRidge : num 0 0 0 0 1 0 0 0 0 0 ...  
## $ NeighborhoodNridgHt : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodNWAmes : num 0 0 0 0 0 0 0 1 0 0 ...  
## $ NeighborhoodOldTown : num 0 0 0 0 0 0 0 0 1 0 ...  
## $ NeighborhoodSawyer : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodSawyerW : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodSomerst : num 0 0 0 0 0 0 1 0 0 0 ...  
## $ NeighborhoodStoneBr : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodSWISU : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodTimber : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodVeenker : num 0 1 0 0 0 0 0 0 0 0 ...  
## $ Condition1Feedr : num 0 1 0 0 0 0 0 0 0 0 ...  
## $ Condition1Norm : num 1 0 1 1 1 1 1 0 0 0 ...  
## $ Condition1PosN : num 0 0 0 0 0 0 0 1 0 0 ...  
## $ Condition1RRAe : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Condition1RRAn : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Condition2Norm : num 1 1 1 1 1 1 1 1 1 0 ...  
## $ BldgType2fmCon : num 0 0 0 0 0 0 0 0 0 1 ...  
## $ BldgTypeDuplex : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ BldgTypeTwnhs : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ BldgTypeTwnhsE : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ HouseStyle1.5Unf : num 0 0 0 0 0 0 0 0 0 1 ...  
## $ HouseStyle1Story : num 0 1 0 0 0 0 1 0 0 0 ...  
## $ HouseStyle2.5Unf : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ HouseStyle2Story : num 1 0 1 1 1 0 0 1 0 0 ...  
## $ HouseStyleSFoyer : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ HouseStyleSLvl : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ OverallQual : num 7 6 7 7 8 5 8 7 7 5 ...  
## $ OverallCond : num 5 8 5 5 5 5 5 6 5 6 ...  
## [list output truncated]

View(train.set)  
test.set <- df.combined.predictors[is.na(df.combined$SalePrice), ]  
str(test.set)

## 'data.frame': 1459 obs. of 182 variables:  
## $ LotFrontage : num 0.575 0.613 0.34 0.499 -1.287 ...  
## $ LotArea : num 0.526 0.929 0.868 0.225 -1.133 ...  
## $ YearBuilt : num -0.34 -0.439 0.849 0.882 0.684 ...  
## $ MasVnrArea : num -0.793 0.999 -0.793 0.37 -0.793 ...  
## $ BsmtFinSF1 : num 0.647 0.874 0.823 0.731 0.454 ...  
## $ BsmtFinSF2 : num 2.286 -0.363 -0.363 -0.363 -0.363 ...  
## $ BsmtUnfSF : num -0.012 0.2055 -0.373 0.0851 0.6958 ...  
## $ X1stFlrSF : num -0.636 0.591 -0.527 -0.534 0.474 ...  
## $ X2ndFlrSF : num -0.863 -0.863 1.138 1.128 -0.863 ...  
## $ LowQualFinSF : num -0.117 -0.117 -0.117 -0.117 -0.117 ...  
## $ GrLivArea : num -1.429 -0.209 0.42 0.372 -0.326 ...  
## $ BsmtFullBath : num -0.819 -0.819 -0.819 -0.819 -0.819 ...  
## $ BsmtHalfBath : num -0.252 -0.252 -0.252 -0.252 -0.252 ...  
## $ FullBath : num -1.027 -1.027 0.783 0.783 0.783 ...  
## $ HalfBath : num -0.755 1.233 1.233 1.233 -0.755 ...  
## $ BedroomAbvGr : num -1.05 0.17 0.17 0.17 -1.05 ...  
## $ KitchenAbvGr : num -0.201 -0.201 -0.201 -0.201 -0.201 ...  
## $ Fireplaces : num -0.925 -0.925 0.626 0.626 -0.925 ...  
## $ GarageCars : num -1.004 -1.004 0.308 0.308 0.308 ...  
## $ WoodDeckSF : num 0.965 1.361 1.124 1.328 -0.944 ...  
## $ OpenPorchSF : num -1.082 0.595 0.569 0.595 0.97 ...  
## $ EnclosedPorch : num -0.428 -0.428 -0.428 -0.428 -0.428 ...  
## $ X3SsnPorch : num -0.113 -0.113 -0.113 -0.113 -0.113 ...  
## $ ScreenPorch : num 2.975 -0.309 -0.309 -0.309 3.099 ...  
## $ PoolArea : num -0.0641 -0.0641 -0.0641 -0.0641 -0.0641 ...  
## $ MiscVal : num -0.189 7.415 -0.189 -0.189 -0.189 ...  
## $ TotalBath : num -1.508 -0.888 0.351 0.351 -0.269 ...  
## $ Age : num 1.219 1.362 -0.553 -0.553 -0.265 ...  
## $ TotalSqFt : num -1.006 0.287 0.162 0.128 0.166 ...  
## $ TotalPorchSqFt : num 0.74 0.198 0.173 0.198 1.028 ...  
## $ MSSubClass1 Story 1946+ : num 1 1 0 0 0 0 1 0 1 1 ...  
## $ MSSubClass1 Story 1945- : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClass1.5 Story Unf All : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClass1.5 Story Fin All : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClass2 Story 1946+ : num 0 0 1 1 0 1 0 1 0 0 ...  
## $ MSSubClass2 Story 1945- : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClass2.5 Story All : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClassSplit/Multi Level : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClassSplit Foyer : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClassDuplex All Style/Age: num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClass1 Story PUD 1946+ : num 0 0 0 0 1 0 0 0 0 0 ...  
## $ MSSubClass2 Story PUD 1946+ : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClassPUD Multilevel : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSSubClass2 Family Conversion : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSZoningFV : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ MSZoningRH : num 1 0 0 0 0 0 0 0 0 0 ...  
## $ MSZoningRL : num 0 1 1 1 1 1 1 1 1 1 ...  
## $ MSZoningRM : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Street : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ AlleyNone : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ AlleyPave : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ LotShape : num 3 2 2 2 2 2 2 2 3 3 ...  
## $ LandContourHLS : num 0 0 0 0 1 0 0 0 0 0 ...  
## $ LandContourLow : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ LandContourLvl : num 1 1 1 1 0 1 1 1 1 1 ...  
## $ LotConfigCulDSac : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ LotConfigFR2 : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ LotConfigInside : num 1 0 1 1 1 0 1 1 1 0 ...  
## $ LandSlope : num 2 2 2 2 2 2 2 2 2 2 ...  
## $ NeighborhoodBrDale : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodBrkSide : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodClearCr : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodCollgCr : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodCrawfor : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodEdwards : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodGilbert : num 0 0 1 1 0 1 1 1 1 0 ...  
## $ NeighborhoodIDOTRR : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodMeadowV : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodMitchel : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodNAmes : num 1 1 0 0 0 0 0 0 0 1 ...  
## $ NeighborhoodNoRidge : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodNridgHt : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodNWAmes : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodOldTown : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodSawyer : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodSawyerW : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodSomerst : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodStoneBr : num 0 0 0 0 1 0 0 0 0 0 ...  
## $ NeighborhoodSWISU : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodTimber : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ NeighborhoodVeenker : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Condition1Feedr : num 1 0 0 0 0 0 0 0 0 0 ...  
## $ Condition1Norm : num 0 1 1 1 1 1 1 1 1 1 ...  
## $ Condition1PosN : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Condition1RRAe : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Condition1RRAn : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Condition2Norm : num 1 1 1 1 1 1 1 1 1 1 ...  
## $ BldgType2fmCon : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ BldgTypeDuplex : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ BldgTypeTwnhs : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ BldgTypeTwnhsE : num 0 0 0 0 1 0 0 0 0 0 ...  
## $ HouseStyle1.5Unf : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ HouseStyle1Story : num 1 1 0 0 1 0 1 0 1 1 ...  
## $ HouseStyle2.5Unf : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ HouseStyle2Story : num 0 0 1 1 0 1 0 1 0 0 ...  
## $ HouseStyleSFoyer : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ HouseStyleSLvl : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ OverallQual : num 5 6 5 6 8 6 6 6 7 4 ...  
## $ OverallCond : num 6 6 5 6 5 5 7 5 5 5 ...  
## [list output truncated]

View(test.set)

##Lasso Regresion Model  
set.seed(12345678)  
Control.Train <-trainControl(method = "cv", number = 5)  
Grid.Lasso <- expand.grid(alpha = 1, lambda = seq(0.001, 0.1, by = 0.0005))  
  
Model.Lasso <- train(x = train.set, y = df.combined$SalePrice[!is.na(df.combined$SalePrice)], method = 'glmnet', trControl= Control.Train, tuneGrid = Grid.Lasso)  
Model.Lasso$bestTune

## alpha lambda  
## 5 1 0.003

min(Model.Lasso$results$RMSE)

## [1] 0.1142001

Vars.Important.Lasso <- varImp(Model.Lasso, scale = F)  
Importance.Lasso <- Vars.Important.Lasso$importance  
  
Vars.Selected.Lasso <- length(which(Importance.Lasso$Overall!= 0))  
Vars.NotSelected.Lasso <- length(which(Importance.Lasso$Overall == 0))  
  
cat('Lasso Model used', Vars.Selected.Lasso, 'variables & did not use', Vars.NotSelected.Lasso)

## Lasso Model used 85 variables & did not use 97

Prediction.Lasso <- predict(Model.Lasso, test.set)  
Prediction.Values.Lasso <- exp(Prediction.Lasso)  
  
View(Prediction.Values.Lasso)  
summary(Prediction.Values.Lasso)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 47844 126908 158759 178107 212839 685017

##XGBoost Modeling  
Grid.XGB <- expand.grid(  
 nrounds = 1000,  
 eta = c(0.1, 0.05, 0.01),  
 max\_depth = c(2, 3, 4, 5, 6),  
 gamma = 0,  
 colsample\_bytree = 1,  
 min\_child\_weight = c(1, 2, 3, 4, 5),  
 subsample = 1  
)  
  
##Find Best Hyperparameter Values  
Caret.XGB <- train(x = train.set, y = df.combined$SalePrice[!is.na(df.combined$SalePrice)], method = 'xgbTree', trControl = Control.Train, tuneGrid = Grid.XGB)  
Caret.XGB$bestTune

## nrounds max\_depth eta gamma colsample\_bytree min\_child\_weight  
## 33 1000 3 0.05 0 1 3  
## subsample  
## 33 1

##XGB Boost Train & Test Sets  
Label.Train.XGB <- df.combined$SalePrice[!is.na(df.combined$SalePrice)]  
  
##XBG Train & Test Sets in Matrix  
Train.Matrix.XGB <- xgb.DMatrix(data = as.matrix(train.set), label = Label.Train.XGB)  
Test.Matrix.XGB <- xgb.DMatrix(data = as.matrix(test.set))  
  
##XGB Parameters  
Parameters.Model.XGB <- list(  
 objective = "reg:linear",  
 booster = "gbtree",  
 eta = 0.05,  
 gamma = 0,  
 max\_depth = 3,  
 min\_child\_weight = 3,  
 subsample = 1,  
 colsample\_bytree = 1  
)  
  
##XGB Cross Validation  
Cross.Validation.XGB <- xgb.cv(params = Parameters.Model.XGB, data = Train.Matrix.XGB, nrounds = 500, nfold = 5, showsd = T, stratified = T, print\_every\_n = 40, early\_stopping\_rounds = 10, maximize = F)

## [1] train-rmse:10.955587+0.004653 test-rmse:10.955522+0.019857   
## Multiple eval metrics are present. Will use test\_rmse for early stopping.  
## Will train until test\_rmse hasn't improved in 10 rounds.  
##   
## [41] train-rmse:1.428223+0.000292 test-rmse:1.428729+0.010585   
## [81] train-rmse:0.219372+0.000623 test-rmse:0.231261+0.004968   
## [121] train-rmse:0.101754+0.001338 test-rmse:0.130095+0.009549   
## [161] train-rmse:0.089498+0.001712 test-rmse:0.123474+0.010629   
## [201] train-rmse:0.083547+0.001728 test-rmse:0.121148+0.010802   
## [241] train-rmse:0.079052+0.001580 test-rmse:0.119955+0.011079   
## [281] train-rmse:0.075318+0.001398 test-rmse:0.119487+0.010939   
## [321] train-rmse:0.072064+0.001312 test-rmse:0.119232+0.011073   
## [361] train-rmse:0.069035+0.001125 test-rmse:0.118974+0.011057   
## Stopping. Best iteration:  
## [367] train-rmse:0.068597+0.001119 test-rmse:0.118900+0.011069

##Train Model using Best Round from Cross Validation  
Model.XGB <- xgb.train(data = Train.Matrix.XGB, params = Parameters.Model.XGB, nrounds = 367)  
  
Prediction.XGB <- predict(Model.XGB, Test.Matrix.XGB)  
Prediction.Values.XGB <- exp(Prediction.XGB)  
  
head(Prediction.Values.XGB)

## [1] 118315.0 161907.8 186297.7 187056.5 193928.6 166106.4

View(Prediction.Values.XGB)  
summary(Prediction.Values.XGB)

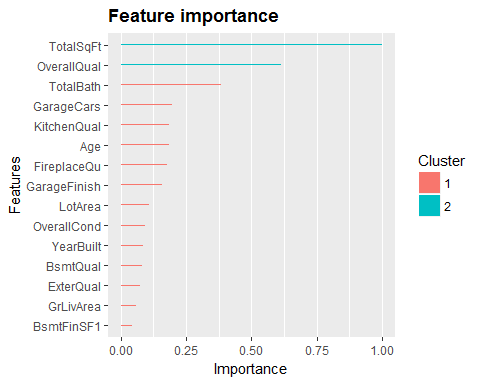
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 42644 127825 156311 179084 209042 675123

##KMeans Clustering identifying Importance of Variables  
install.packages("Ckmeans.1d.dp")

## Installing package into 'C:/Users/cizek/Documents/R/win-library/3.4'  
## (as 'lib' is unspecified)

## package 'Ckmeans.1d.dp' successfully unpacked and MD5 sums checked  
##   
## The downloaded binary packages are in  
## C:\Users\cizek\AppData\Local\Temp\RtmpSo8y9i\downloaded\_packages

library(Ckmeans.1d.dp)  
  
Importance.XGB <- xgb.importance(feature\_names = colnames(train.set), Model.XGB)  
xgb.ggplot.importance(importance\_matrix = Importance.XGB[1:15], rel\_to\_first = TRUE)



##Averaging Models  
Models.Average <- data.frame(Id = test.IDs, SalePrice = (Prediction.Values.XGB + Prediction.Values.Lasso)/2)  
  
head(Models.Average)

## Id SalePrice  
## 1461 1461 116905.4  
## 1462 1462 161620.0  
## 1463 1463 182991.7  
## 1464 1464 192431.8  
## 1465 1465 200390.8  
## 1466 1466 167897.1

View(Models.Average)  
summary(Models.Average$SalePrice)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 46871 127287 156887 178595 209627 680070

##Export Results  
write.csv(Models.Average, file = 'Capstone Results.csv', row.names = F)