House Price Prediction

Jung Hyun Kim

12/23/2021

Data Cleaning

```
train<- read.csv("train.csv")
test <- read.csv("test.csv")

dim(train)
```

```
## [1] 1460   81
```

```
dim(test)
```

```
## [1] 1459   80
```

```
#Combining train and test for data cleaning purpose
which(!colnames(train)%in%colnames(test))
```

```
## [1] 81
```

```
colnames(train)[81]
```

```
## [1] "SalePrice"
```

```
# because test is lacking 2 columns
test$SalePrice <- 0

# Data Cleaning
data <- data.frame(rbind(train,test))

# chr to factor
```

The output above shows the number of missing observations("NA") for each variables in the dataset.

LotFrontage is a variable for linear feet of street connected to property, which indicates a home's accessibility. The value NA could mean either missing value or literally no access to a street, which sounds illogical given that a fact that the land that houses were built on is owned by the home owners.

Evaluating Lot Frontage

```r
summary(data$LotFrontage)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   21.00   59.00   68.00   69.31   80.00  313.00     486
```

The 5 number summary above shows that NA's are more likely to be missing values. For this case, KNN is used to replace NA's.

```r
library("VIM")

data <- kNN(data, variable= "LotFrontage", k=5, imp_var=FALSE, imp_suffix = NULL)
```

Alley can be left as it is based on the data description.

```r
data[which(is.na(data$Alley)),"Alley"] <- "None"
```

NA for "MasVnrType" is replaced with "None"

```r
data[which(is.na(data$MasVnrType)), "MasVnrType"] <- "None"
```

For Exterior1st, it's hard to replace NA since levels are categorical without anything ambiguous like "None". Hence, the NA row is removed. This is reasonable because it's only 1 observation, and the variable might be useful(probably not for regression because of too many levels).

Same approach was taken for Exterior2nd

```r
c(which(is.na(data$Exterior1st)),which(is.na(data$Exterior2nd))) # same row 2152
```

```
## [1] 2152 2152
```

```r
data <- data[-which(is.na(data$Exterior1st)),]
```

For all other variables, if missing value is fewer than 5, all of them will be removed

```r
colSums(is.na(data)) # current na obs. for each variables
```

```
##           Id   MSSubClass     MSZoning  LotFrontage      LotArea
##            0            0            4            0            0
##       Street        Alley     LotShape  LandContour    Utilities
##            0            0            0            0            2
##    LotConfig    LandSlope Neighborhood   Condition1   Condition2
##            0            0            0            0            0
##     BldgType   HouseStyle  OverallQual  OverallCond    YearBuilt
##            0            0            0            0            0
## YearRemodAdd    RoofStyle     RoofMatl  Exterior1st  Exterior2nd
##            0            0            0            0            0
##   MasVnrType   MasVnrArea    ExterQual    ExterCond   Foundation
##            0           23            0            0            0
##     BsmtQual     BsmtCond BsmtExposure BsmtFinType1   BsmtFinSF1
```

```
##            81           82           82           79            1
##  BsmtFinType2    BsmtFinSF2    BsmtUnfSF  TotalBsmtSF       Heating
##            80            1            1            1             0
##     HeatingQC    CentralAir   Electrical     X1stFlrSF     X2ndFlrSF
##             0            0            1            0             0
##  LowQualFinSF     GrLivArea  BsmtFullBath BsmtHalfBath      FullBath
##             0            0            2            2             0
##      HalfBath  BedroomAbvGr  KitchenAbvGr   KitchenQual  TotRmsAbvGrd
##             0            0            0            1             0
##    Functional    Fireplaces   FireplaceQu    GarageType   GarageYrBlt
##             2            0         1420          156           158
##  GarageFinish    GarageCars    GarageArea    GarageQual    GarageCond
##           158            1            1          158           158
##    PavedDrive    WoodDeckSF   OpenPorchSF EnclosedPorch    X3SsnPorch
##             0            0            0            0             0
##   ScreenPorch      PoolArea        PoolQC         Fence   MiscFeature
##             0            0         2908         2347          2813
##       MiscVal        MoSold        YrSold      SaleType SaleCondition
##             0            0            0            1             0
##     SalePrice
##             0
```

```
varnames <- names(data)[(colSums(is.na(data))>=1 &colSums(is.na(data))<=5)] # any variables =1 and less

#replacing na's with KNN
for(i in varnames){
  a <-which(is.na(data[,i]))
  data <- kNN(data, variable= i, k=5,imp_var=FALSE, imp_suffix = NULL)
}
```

MasVnrArea should be replaced with KNN since it's Masonry veneer area in square feet.

```
data <- kNN(data,"MasVnrArea",k=5,imp_var=FALSE, imp_suffix = NULL)
```

All remaining variables are replaced with kNN

```
varnames2 <- names(data)[colSums(is.na(data))>0]

for ( i in varnames2){
  data <- kNN(data,i,k=5,imp_var=FALSE, imp_suffix = NULL)
}
```

Converting character variables to factors

```
data[colnames(Filter(is.character,(data)))] <-lapply(data[colnames(Filter(is.character,(data)))],factor
```

simple OLS

```r
data <- data[,-1] #removing ID column as it is unecessary
```

```r
ols <- lm(SalePrice~., data = data[1:nrow(train),])
summary(ols)
```

```
##
## Call:
## lm(formula = SalePrice ~ ., data = data[1:nrow(train), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -179023   -9591     226    9905  179023
##
## Coefficients: (3 not defined because of singularities)
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -5.046e+05  1.054e+06  -0.479 0.632344
## MSSubClass           -5.792e+01  8.312e+01  -0.697 0.486020
## MSZoningFV            3.271e+04  1.201e+04   2.724 0.006536 **
## MSZoningRH            2.486e+04  1.190e+04   2.090 0.036856 *
## MSZoningRL            2.531e+04  1.020e+04   2.481 0.013250 *
## MSZoningRM            2.178e+04  9.574e+03   2.275 0.023106 *
## LotFrontage           4.732e+01  4.615e+01   1.025 0.305387
## LotArea               7.400e-01  1.095e-01   6.755 2.21e-11 ***
## StreetPave            3.074e+04  1.211e+04   2.538 0.011258 *
## AlleyNone            -7.540e+02  4.242e+03  -0.178 0.858930
## AlleyPave            -6.096e+02  6.057e+03  -0.101 0.919856
## LotShapeIR2           5.275e+03  4.263e+03   1.237 0.216224
## LotShapeIR3           3.590e+03  8.908e+03   0.403 0.687034
## LotShapeReg           1.615e+03  1.609e+03   1.004 0.315725
## LandContourHLS        9.045e+03  5.122e+03   1.766 0.077685 .
## LandContourLow       -9.412e+03  6.361e+03  -1.480 0.139249
## LandContourLvl        6.306e+03  3.700e+03   1.704 0.088605 .
## UtilitiesNoSeWa      -3.673e+04  2.639e+04  -1.392 0.164270
## LotConfigCulDSac      8.762e+03  3.361e+03   2.607 0.009240 **
## LotConfigFR2         -7.406e+03  4.077e+03  -1.817 0.069518 .
## LotConfigFR3         -1.516e+04  1.260e+04  -1.203 0.229018
## LotConfigInside      -1.106e+03  1.805e+03  -0.613 0.540225
## LandSlopeMod          6.351e+03  3.987e+03   1.593 0.111417
## LandSlopeSev         -4.377e+04  1.144e+04  -3.827 0.000136 ***
## NeighborhoodBlueste   5.014e+03  1.910e+04   0.263 0.792948
## NeighborhoodBrDale    1.146e+03  1.110e+04   0.103 0.917777
## NeighborhoodBrkSide  -4.568e+03  9.538e+03  -0.479 0.632089
## NeighborhoodClearCr  -1.387e+04  9.196e+03  -1.508 0.131822
## NeighborhoodCollgCr  -9.642e+03  7.283e+03  -1.324 0.185778
## NeighborhoodCrawfor   1.163e+04  8.590e+03   1.354 0.175911
## NeighborhoodEdwards  -2.004e+04  8.049e+03  -2.490 0.012920 *
## NeighborhoodGilbert  -1.255e+04  7.706e+03  -1.629 0.103572
## NeighborhoodIDOTRR   -1.072e+04  1.077e+04  -0.996 0.319541
## NeighborhoodMeadowV  -3.214e+03  1.132e+04  -0.284 0.776513
## NeighborhoodMitchel  -2.093e+04  8.220e+03  -2.546 0.011023 *
## NeighborhoodNAmes    -1.581e+04  7.880e+03  -2.006 0.045058 *
```

```
## NeighborhoodNoRidge   2.730e+04  8.467e+03   3.224 0.001298 **
## NeighborhoodNPkVill   1.465e+04  1.406e+04   1.042 0.297794
## NeighborhoodNridgHt   1.907e+04  7.542e+03   2.529 0.011569 *
## NeighborhoodNWAmes   -1.836e+04  8.085e+03  -2.271 0.023296 *
## NeighborhoodOldTown  -1.309e+04  9.716e+03  -1.347 0.178193
## NeighborhoodSawyer   -1.085e+04  8.169e+03  -1.328 0.184503
## NeighborhoodSawyerW  -3.858e+03  7.822e+03  -0.493 0.621919
## NeighborhoodSomerst  -1.739e+03  9.049e+03  -0.192 0.847622
## NeighborhoodStoneBr   3.866e+04  8.329e+03   4.642 3.83e-06 ***
## NeighborhoodSWISU    -9.566e+03  9.739e+03  -0.982 0.326157
## NeighborhoodTimber   -9.029e+03  8.150e+03  -1.108 0.268188
## NeighborhoodVeenker  -8.908e+01  1.054e+04  -0.008 0.993257
## Condition1Feedr       7.636e+03  5.017e+03   1.522 0.128241
## Condition1Norm        1.572e+04  4.175e+03   3.766 0.000174 ***
## Condition1PosA        6.580e+03  1.003e+04   0.656 0.511768
## Condition1PosN        1.325e+04  7.445e+03   1.780 0.075382 .
## Condition1RRAe       -1.659e+04  9.118e+03  -1.819 0.069153 .
## Condition1RRAn        1.194e+04  6.956e+03   1.716 0.086384 .
## Condition1RRNe        4.590e+00  1.755e+04   0.000 0.999791
## Condition1RRNn        1.002e+04  1.289e+04   0.778 0.436861
## Condition2Feedr      -9.335e+03  2.354e+04  -0.397 0.691706
## Condition2Norm       -1.043e+04  2.036e+04  -0.512 0.608665
## Condition2PosA        3.213e+04  3.719e+04   0.864 0.387751
## Condition2PosN       -2.394e+05  2.776e+04  -8.626  < 2e-16 ***
## Condition2RRAe       -1.127e+05  5.716e+04  -1.971 0.048943 *
## Condition2RRAn       -2.427e+04  3.165e+04  -0.767 0.443376
## Condition2RRNn       -3.292e+03  2.717e+04  -0.121 0.903580
## BldgType2fmCon       -9.199e+02  1.253e+04  -0.073 0.941503
## BldgTypeDuplex       -6.274e+03  7.411e+03  -0.847 0.397367
## BldgTypeTwnhs        -1.966e+04  1.006e+04  -1.954 0.050907 .
## BldgTypeTwnhsE       -1.564e+04  9.084e+03  -1.722 0.085284 .
## HouseStyle1.5Unf      1.509e+04  7.902e+03   1.910 0.056416 .
## HouseStyle1Story      8.502e+03  4.356e+03   1.952 0.051176 .
## HouseStyle2.5Fin     -2.285e+04  1.229e+04  -1.859 0.063222 .
## HouseStyle2.5Unf     -8.811e+03  9.249e+03  -0.953 0.340968
## HouseStyle2Story     -5.210e+03  3.522e+03  -1.479 0.139372
## HouseStyleSFoyer      3.528e+03  6.262e+03   0.563 0.573211
## HouseStyleSLvl        4.808e+03  5.517e+03   0.871 0.383696
## OverallQual           6.424e+03  1.015e+03   6.328 3.48e-10 ***
## OverallCond           5.478e+03  8.759e+02   6.254 5.53e-10 ***
## YearBuilt             3.386e+02  7.671e+01   4.414 1.11e-05 ***
## YearRemodAdd          1.105e+02  5.625e+01   1.964 0.049793 *
## RoofStyleGable        9.670e+03  1.845e+04   0.524 0.600217
## RoofStyleGambrel      9.717e+03  2.019e+04   0.481 0.630413
## RoofStyleHip          8.944e+03  1.851e+04   0.483 0.629038
## RoofStyleMansard      2.037e+04  2.144e+04   0.950 0.342166
## RoofStyleShed         9.911e+04  3.471e+04   2.855 0.004372 **
## RoofMatlCompShg       6.760e+05  3.312e+04  20.410  < 2e-16 ***
## RoofMatlMembran       7.748e+05  4.771e+04  16.240  < 2e-16 ***
## RoofMatlMetal         7.443e+05  4.671e+04  15.936  < 2e-16 ***
## RoofMatlRoll          6.613e+05  4.157e+04  15.909  < 2e-16 ***
## RoofMatlTar&Grv       6.815e+05  3.792e+04  17.971  < 2e-16 ***
## RoofMatlWdShake       6.663e+05  3.661e+04  18.201  < 2e-16 ***
## RoofMatlWdShngl       7.312e+05  3.452e+04  21.179  < 2e-16 ***
```

```
## Exterior1stAsphShn   -1.829e+04  3.300e+04  -0.554 0.579562
## Exterior1stBrkComm   -1.088e+04  2.779e+04  -0.391 0.695572
## Exterior1stBrkFace    2.898e+03  1.272e+04   0.228 0.819845
## Exterior1stCBlock    -1.438e+04  2.743e+04  -0.524 0.600292
## Exterior1stCemntBd   -1.507e+04  1.911e+04  -0.789 0.430296
## Exterior1stHdBoard   -2.046e+04  1.291e+04  -1.584 0.113481
## Exterior1stImStucc   -4.427e+04  2.764e+04  -1.601 0.109587
## Exterior1stMetalSd   -1.050e+04  1.460e+04  -0.719 0.472255
## Exterior1stPlywood   -2.116e+04  1.275e+04  -1.660 0.097246 .
## Exterior1stStone     -1.508e+04  2.398e+04  -0.629 0.529628
## Exterior1stStucco    -1.015e+04  1.413e+04  -0.718 0.472808
## Exterior1stVinylSd   -2.020e+04  1.335e+04  -1.513 0.130420
## Exterior1stWd Sdng   -1.802e+04  1.232e+04  -1.463 0.143768
## Exterior1stWdShing   -1.338e+04  1.337e+04  -1.001 0.317098
## Exterior2ndAsphShn    1.869e+04  2.230e+04   0.838 0.401960
## Exterior2ndBrk Cmn    1.101e+04  2.011e+04   0.547 0.584273
## Exterior2ndBrkFace    8.695e+03  1.314e+04   0.662 0.508183
## Exterior2ndCBlock            NA         NA      NA       NA
## Exterior2ndCmentBd    1.405e+04  1.881e+04   0.747 0.455287
## Exterior2ndHdBoard    1.452e+04  1.242e+04   1.169 0.242442
## Exterior2ndImStucc    2.912e+04  1.423e+04   2.046 0.040929 *
## Exterior2ndMetalSd    8.883e+03  1.421e+04   0.625 0.531999
## Exterior2ndOther     -1.041e+04  2.724e+04  -0.382 0.702540
## Exterior2ndPlywood    1.168e+04  1.206e+04   0.969 0.332793
## Exterior2ndStone     -4.109e+03  1.703e+04  -0.241 0.809348
## Exterior2ndStucco     9.510e+03  1.365e+04   0.697 0.486014
## Exterior2ndVinylSd    1.839e+04  1.284e+04   1.432 0.152327
## Exterior2ndWd Sdng    1.443e+04  1.192e+04   1.211 0.226205
## Exterior2ndWd Shng    8.943e+03  1.248e+04   0.717 0.473800
## MasVnrTypeBrkFace     6.592e+03  6.877e+03   0.959 0.337982
## MasVnrTypeNone        9.541e+03  6.911e+03   1.381 0.167673
## MasVnrTypeStone       1.226e+04  7.249e+03   1.691 0.091102 .
## MasVnrArea            1.966e+01  5.793e+00   3.394 0.000712 ***
## ExterQualFa          -9.356e+03  1.116e+04  -0.838 0.402175
## ExterQualGd          -2.007e+04  4.809e+03  -4.174 3.21e-05 ***
## ExterQualTA          -2.121e+04  5.319e+03  -3.988 7.05e-05 ***
## ExterCondFa          -6.818e+03  1.814e+04  -0.376 0.707035
## ExterCondGd          -1.113e+04  1.727e+04  -0.645 0.519250
## ExterCondPo           4.793e+03  3.147e+04   0.152 0.878972
## ExterCondTA          -7.915e+03  1.724e+04  -0.459 0.646297
## FoundationCBlock      2.656e+03  3.198e+03   0.830 0.406561
## FoundationPConc       4.376e+03  3.428e+03   1.277 0.201989
## FoundationSlab        3.395e+03  7.783e+03   0.436 0.662828
## FoundationStone       6.076e+03  1.130e+04   0.538 0.590918
## FoundationWood       -3.145e+04  1.481e+04  -2.123 0.033969 *
## BsmtQualFa           -1.259e+04  6.361e+03  -1.980 0.047971 *
## BsmtQualGd           -1.826e+04  3.344e+03  -5.459 5.80e-08 ***
## BsmtQualTA           -1.487e+04  4.162e+03  -3.573 0.000366 ***
## BsmtCondGd            8.096e+02  5.278e+03   0.153 0.878109
## BsmtCondPo            7.367e+04  2.988e+04   2.465 0.013825 *
## BsmtCondTA            3.538e+03  4.247e+03   0.833 0.404944
## BsmtExposureGd        1.439e+04  2.972e+03   4.841 1.46e-06 ***
## BsmtExposureMn       -3.780e+03  3.021e+03  -1.251 0.211133
## BsmtExposureNo       -5.259e+03  2.173e+03  -2.420 0.015672 *
```

```
## BsmtFinType1BLQ       3.472e+03  2.829e+03   1.227 0.219899
## BsmtFinType1GLQ       5.777e+03  2.524e+03   2.289 0.022250 *
## BsmtFinType1LwQ      -3.288e+03  3.747e+03  -0.877 0.380449
## BsmtFinType1Rec       1.911e+01  3.010e+03   0.006 0.994937
## BsmtFinType1Unf       4.205e+03  2.871e+03   1.465 0.143210
## BsmtFinSF1            3.492e+01  4.569e+00   7.643 4.30e-14 ***
## BsmtFinType2BLQ      -1.248e+04  7.580e+03  -1.647 0.099821 .
## BsmtFinType2GLQ      -2.979e+03  9.349e+03  -0.319 0.750097
## BsmtFinType2LwQ      -1.529e+04  7.415e+03  -2.062 0.039424 *
## BsmtFinType2Rec      -1.145e+04  7.120e+03  -1.608 0.108147
## BsmtFinType2Unf      -1.006e+04  7.553e+03  -1.332 0.183253
## BsmtFinSF2            2.564e+01  8.540e+00   3.002 0.002733 **
## BsmtUnfSF             1.544e+01  4.021e+00   3.841 0.000129 ***
## TotalBsmtSF                  NA         NA      NA       NA
## HeatingGasA          -3.196e+03  2.469e+04  -0.129 0.897011
## HeatingGasW          -7.573e+03  2.556e+04  -0.296 0.767105
## HeatingGrav          -1.036e+04  2.694e+04  -0.385 0.700579
## HeatingOthW          -2.280e+04  3.105e+04  -0.734 0.463035
## HeatingWall           9.148e+03  2.864e+04   0.319 0.749517
## HeatingQCFa           8.175e+02  4.729e+03   0.173 0.862769
## HeatingQCGd          -3.842e+03  2.076e+03  -1.851 0.064441 .
## HeatingQCPo           2.580e+03  2.671e+04   0.097 0.923062
## HeatingQCTA          -3.571e+03  2.083e+03  -1.715 0.086679 .
## CentralAirY          -3.092e+02  3.892e+03  -0.079 0.936684
## ElectricalFuseF       5.781e+02  5.783e+03   0.100 0.920392
## ElectricalFuseP      -5.825e+03  1.859e+04  -0.313 0.754038
## ElectricalMix        -4.998e+04  4.466e+04  -1.119 0.263339
## ElectricalSBrkr      -1.232e+03  2.960e+03  -0.416 0.677280
## X1stFlrSF             4.964e+01  5.251e+00   9.453  < 2e-16 ***
## X2ndFlrSF             6.801e+01  5.576e+00  12.196  < 2e-16 ***
## LowQualFinSF          1.185e+01  1.841e+01   0.644 0.519730
## GrLivArea                    NA         NA      NA       NA
## BsmtFullBath          9.773e+02  1.984e+03   0.493 0.622371
## BsmtHalfBath         -8.105e+02  3.029e+03  -0.268 0.789088
## FullBath              3.818e+03  2.209e+03   1.728 0.084155 .
## HalfBath              1.447e+03  2.104e+03   0.688 0.491725
## BedroomAbvGr         -3.705e+03  1.372e+03  -2.700 0.007030 **
## KitchenAbvGr         -1.403e+04  5.729e+03  -2.449 0.014460 *
## KitchenQualFa        -2.078e+04  6.224e+03  -3.338 0.000868 ***
## KitchenQualGd        -2.561e+04  3.487e+03  -7.344 3.79e-13 ***
## KitchenQualTA        -2.435e+04  3.927e+03  -6.201 7.69e-10 ***
## TotRmsAbvGrd          1.469e+03  9.566e+02   1.536 0.124902
## FunctionalMaj2        2.926e+03  1.442e+04   0.203 0.839231
## FunctionalMin1        1.060e+04  8.622e+03   1.229 0.219261
## FunctionalMin2        1.266e+04  8.617e+03   1.469 0.142187
## FunctionalMod        -1.036e+03  1.057e+04  -0.098 0.921915
## FunctionalSev        -3.731e+04  2.944e+04  -1.267 0.205248
## FunctionalTyp         2.247e+04  7.450e+03   3.016 0.002618 **
## Fireplaces            2.158e+03  1.349e+03   1.600 0.109770
## FireplaceQuFa        -1.868e+03  6.120e+03  -0.305 0.760239
## FireplaceQuGd         6.560e+02  5.298e+03   0.124 0.901478
## FireplaceQuPo         6.140e+03  7.002e+03   0.877 0.380727
## FireplaceQuTA         3.271e+03  5.451e+03   0.600 0.548541
## GarageTypeAttchd      1.754e+04  1.102e+04   1.591 0.111938
```

```
## GarageTypeBasment     2.179e+04  1.275e+04   1.709 0.087619 .
## GarageTypeBuiltIn     1.593e+04  1.148e+04   1.388 0.165275
## GarageTypeCarPort     2.033e+04  1.467e+04   1.385 0.166184
## GarageTypeDetchd      2.211e+04  1.103e+04   2.005 0.045231 *
## GarageYrBlt          -1.560e+01  5.710e+01  -0.273 0.784686
## GarageFinishRFn      -1.581e+03  2.011e+03  -0.786 0.432144
## GarageFinishUnf       9.641e+02  2.372e+03   0.406 0.684512
## GarageCars            3.067e+03  2.194e+03   1.398 0.162344
## GarageArea            1.341e+01  7.743e+00   1.733 0.083436 .
## GarageQualFa         -1.194e+05  3.022e+04  -3.950 8.26e-05 ***
## GarageQualGd         -1.102e+05  3.096e+04  -3.559 0.000387 ***
## GarageQualPo         -1.336e+05  3.852e+04  -3.467 0.000545 ***
## GarageQualTA         -1.109e+05  2.993e+04  -3.706 0.000220 ***
## GarageCondFa          1.025e+05  3.476e+04   2.950 0.003239 **
## GarageCondGd          1.018e+05  3.588e+04   2.837 0.004628 **
## GarageCondPo          1.062e+05  3.735e+04   2.843 0.004543 **
## GarageCondTA          1.046e+05  3.448e+04   3.033 0.002474 **
## PavedDriveP          -5.716e+03  5.513e+03  -1.037 0.300039
## PavedDriveY          -2.110e+03  3.440e+03  -0.613 0.539862
## WoodDeckSF            1.406e+01  5.845e+00   2.406 0.016281 *
## OpenPorchSF           3.340e+00  1.152e+01   0.290 0.771935
## EnclosedPorch         2.654e+00  1.245e+01   0.213 0.831136
## X3SsnPorch            3.168e+01  2.241e+01   1.413 0.157787
## ScreenPorch           3.789e+01  1.259e+01   3.010 0.002668 **
## PoolArea              1.069e+02  1.966e+01   5.435 6.61e-08 ***
## PoolQCFa             -5.523e+03  3.738e+03  -1.478 0.139790
## PoolQCGd             -1.503e+02  1.489e+03  -0.101 0.919608
## FenceGdWo             3.026e+03  2.552e+03   1.186 0.235927
## FenceMnPrv            4.643e+03  1.781e+03   2.606 0.009271 **
## FenceMnWw            -3.005e+03  7.107e+03  -0.423 0.672444
## MiscFeatureOthr       5.158e+04  6.422e+04   0.803 0.422079
## MiscFeatureShed       4.345e+04  6.621e+04   0.656 0.511811
## MiscFeatureTenC      -3.698e+04  6.495e+04  -0.569 0.569214
## MiscVal               3.337e+00  4.025e+00   0.829 0.407144
## MoSold               -4.436e+02  2.461e+02  -1.803 0.071657 .
## YrSold               -5.581e+02  5.165e+02  -1.081 0.280056
## SaleTypeCon           2.648e+04  1.766e+04   1.500 0.133990
## SaleTypeConLD         1.614e+04  9.710e+03   1.662 0.096833 .
## SaleTypeConLI         5.022e+03  1.158e+04   0.434 0.664588
## SaleTypeConLw        -8.213e+01  1.224e+04  -0.007 0.994647
## SaleTypeCWD           1.562e+04  1.293e+04   1.208 0.227317
## SaleTypeNew           2.196e+04  1.547e+04   1.419 0.156014
## SaleTypeOth           6.270e+03  1.444e+04   0.434 0.664182
## SaleTypeWD           -5.989e+02  4.189e+03  -0.143 0.886343
## SaleConditionAdjLand  1.074e+04  1.467e+04   0.732 0.464426
## SaleConditionAlloca   2.623e+03  8.604e+03   0.305 0.760494
## SaleConditionFamily  -4.474e+01  6.110e+03  -0.007 0.994159
## SaleConditionNormal   6.004e+03  2.905e+03   2.067 0.038971 *
## SaleConditionPartial -1.481e+03  1.489e+04  -0.099 0.920823
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22730 on 1216 degrees of freedom
## Multiple R-squared:  0.9318, Adjusted R-squared:  0.9182
```

```
## F-statistic: 68.35 on 243 and 1216 DF,  p-value: < 2.2e-16
```

```r
sort(ols$coefficients,decreasing = T )
```

```
##      RoofMatlMembran        RoofMatlMetal      RoofMatlWdShngl
##         7.747790e+05         7.443322e+05         7.311521e+05
##      RoofMatlTar&Grv      RoofMatlCompShg      RoofMatlWdShake
##         6.815260e+05         6.760340e+05         6.663175e+05
##         RoofMatlRoll         GarageCondPo         GarageCondTA
##         6.612738e+05         1.062001e+05         1.045604e+05
##         GarageCondFa         GarageCondGd         RoofStyleShed
##         1.025440e+05         1.017940e+05         9.911050e+04
##           BsmtCondPo      MiscFeatureOthr      MiscFeatureShed
##         7.367102e+04         5.157596e+04         4.344809e+04
##  NeighborhoodStoneBr           MSZoningFV       Condition2PosA
##         3.866111e+04         3.270782e+04         3.213466e+04
##           StreetPave   Exterior2ndImStucc  NeighborhoodNoRidge
##         3.074448e+04         2.912498e+04         2.729775e+04
##          SaleTypeCon           MSZoningRL           MSZoningRH
##         2.647785e+04         2.530522e+04         2.485662e+04
##        FunctionalTyp      GarageTypeDetchd          SaleTypeNew
##         2.246534e+04         2.211274e+04         2.195817e+04
##     GarageTypeBasment           MSZoningRM      RoofStyleMansard
##         2.178929e+04         2.177736e+04         2.037080e+04
##     GarageTypeCarPort   NeighborhoodNridgHt   Exterior2ndAsphShn
##         2.032979e+04         1.907363e+04         1.869394e+04
##    Exterior2ndVinylSd      GarageTypeAttchd        SaleTypeConLD
##         1.838767e+04         1.753501e+04         1.613555e+04
##     GarageTypeBuiltIn       Condition1Norm          SaleTypeCWD
##         1.593481e+04         1.572132e+04         1.562356e+04
##      HouseStyle1.5Unf   NeighborhoodNPkVill   Exterior2ndHdBoard
##         1.508910e+04         1.464658e+04         1.452497e+04
##    Exterior2ndWd Sdng       BsmtExposureGd    Exterior2ndCmentBd
##         1.443252e+04         1.438612e+04         1.404696e+04
##       Condition1PosN        FunctionalMin2       MasVnrTypeStone
##         1.324980e+04         1.265596e+04         1.225729e+04
##       Condition1RRAn    Exterior2ndPlywood   NeighborhoodCrawfor
##         1.193823e+04         1.168358e+04         1.163229e+04
##    Exterior2ndBrk Cmn SaleConditionAdjLand        FunctionalMin1
##         1.100681e+04         1.073633e+04         1.059725e+04
##       Condition1RRNn      RoofStyleGambrel        RoofStyleGable
##         1.002304e+04         9.716750e+03         9.669803e+03
##       MasVnrTypeNone     Exterior2ndStucco          HeatingWall
##         9.540535e+03         9.510083e+03         9.147694e+03
##       LandContourHLS          RoofStyleHip     Exterior2ndWd Shng
##         9.044681e+03         8.943695e+03         8.942515e+03
##    Exterior2ndMetalSd      LotConfigCulDSac    Exterior2ndBrkFace
##         8.882876e+03         8.762312e+03         8.695462e+03
##     HouseStyle1Story       Condition1Feedr      MasVnrTypeBrkFace
##         8.501848e+03         7.636115e+03         6.592121e+03
##       Condition1PosA           OverallQual          LandSlopeMod
##         6.580261e+03         6.423502e+03         6.350732e+03
##       LandContourLvl           SaleTypeOth          FireplaceQuPo
##         6.305895e+03         6.269567e+03         6.139684e+03
```

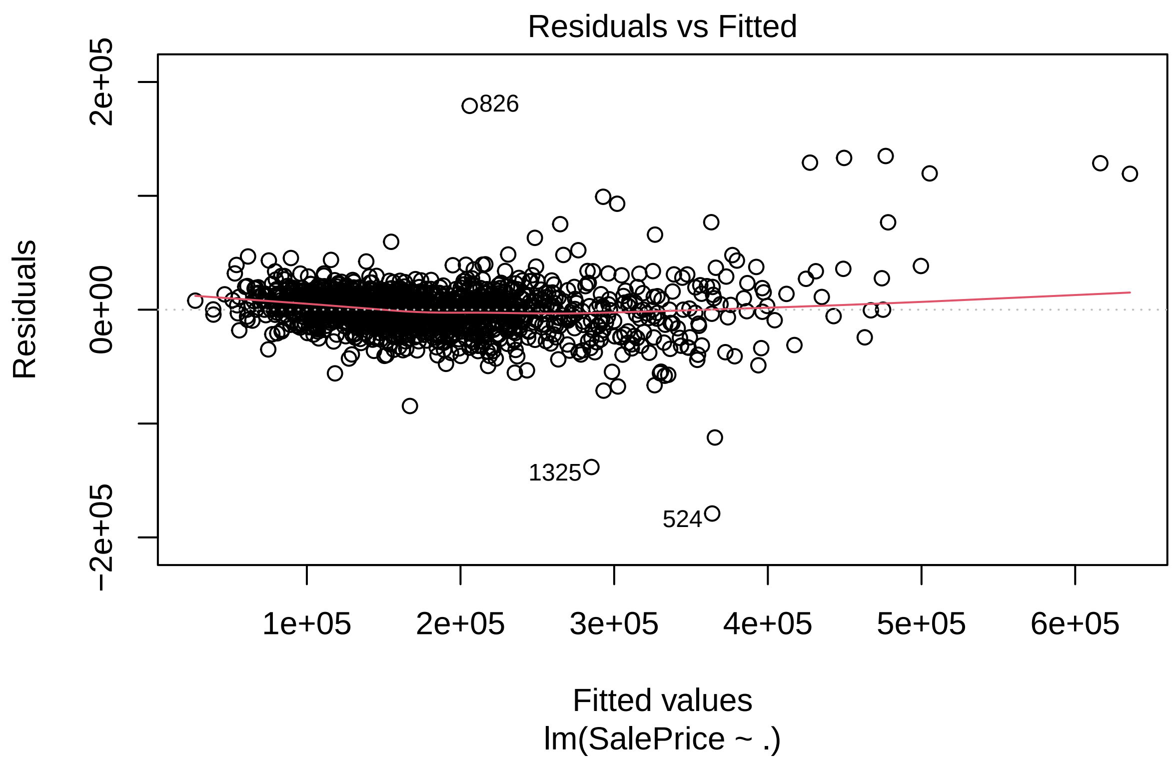
```
##       FoundationStone  SaleConditionNormal       BsmtFinType1GLQ
##          6.075799e+03         6.004306e+03          5.777425e+03
##           OverallCond          LotShapeIR2         SaleTypeConLI
##          5.478118e+03         5.274902e+03          5.021951e+03
##   NeighborhoodBlueste       HouseStyleSLvl          ExterCondPo
##          5.013810e+03         4.807524e+03          4.792754e+03
##            FenceMnPrv      FoundationPConc       BsmtFinType1Unf
##          4.642698e+03         4.376097e+03          4.205267e+03
##              FullBath          LotShapeIR3            BsmtCondTA
##          3.817765e+03         3.589552e+03          3.538322e+03
##      HouseStyleSFoyer      BsmtFinType1BLQ        FoundationSlab
##          3.528279e+03         3.472409e+03          3.394512e+03
##         FireplaceQuTA           GarageCars            FenceGdWo
##          3.271117e+03         3.067031e+03          3.025744e+03
##        FunctionalMaj2  Exterior1stBrkFace     FoundationCBlock
##          2.925794e+03         2.898275e+03          2.655531e+03
##   SaleConditionAlloca          HeatingQCPo            Fireplaces
##          2.623357e+03         2.579780e+03          2.158336e+03
##           LotShapeReg         TotRmsAbvGrd              HalfBath
##          1.614762e+03         1.468953e+03          1.446935e+03
##    NeighborhoodBrDale         BsmtFullBath      GarageFinishUnf
##          1.145797e+03         9.772771e+02          9.640696e+02
##           HeatingQCFa           BsmtCondGd         FireplaceQuGd
##          8.175186e+02         8.096131e+02          6.560086e+02
##       ElectricalFuseF            YearBuilt          YearRemodAdd
##          5.780627e+02         3.385627e+02          1.104603e+02
##              PoolArea            X2ndFlrSF             X1stFlrSF
##          1.068695e+02         6.800895e+01          4.963834e+01
##           LotFrontage          ScreenPorch            BsmtFinSF1
##          4.732231e+01         3.789326e+01          3.492306e+01
##            X3SsnPorch           BsmtFinSF2            MasVnrArea
##          3.168060e+01         2.563920e+01          1.965930e+01
##       BsmtFinType1Rec            BsmtUnfSF            WoodDeckSF
##          1.910497e+01         1.544414e+01          1.406175e+01
##            GarageArea         LowQualFinSF       Condition1RRNe
##          1.341470e+01         1.185187e+01          4.589532e+00
##           OpenPorchSF              MiscVal         EnclosedPorch
##          3.339552e+00         3.337299e+00          2.654449e+00
##               LotArea          GarageYrBlt  SaleConditionFamily
##          7.399740e-01        -1.560352e+01         -4.473820e+01
##            MSSubClass        SaleTypeConLw  NeighborhoodVeenker
##         -5.792442e+01        -8.213284e+01         -8.907527e+01
##              PoolQCGd          CentralAirY                MoSold
##         -1.502743e+02        -3.091986e+02         -4.435985e+02
##                YrSold           SaleTypeWD             AlleyPave
##         -5.581303e+02        -5.989040e+02         -6.095976e+02
##             AlleyNone          BsmtHalfBath        BldgType2fmCon
##         -7.540334e+02        -8.105091e+02         -9.198953e+02
##         FunctionalMod      LotConfigInside       ElectricalSBrkr
##         -1.036374e+03        -1.105638e+03         -1.232206e+03
## SaleConditionPartial      GarageFinishRFn  NeighborhoodSomerst
##         -1.480585e+03        -1.580580e+03         -1.739210e+03
##         FireplaceQuFa          PavedDriveY       BsmtFinType2GLQ
##         -1.868001e+03        -2.109678e+03         -2.978562e+03
```

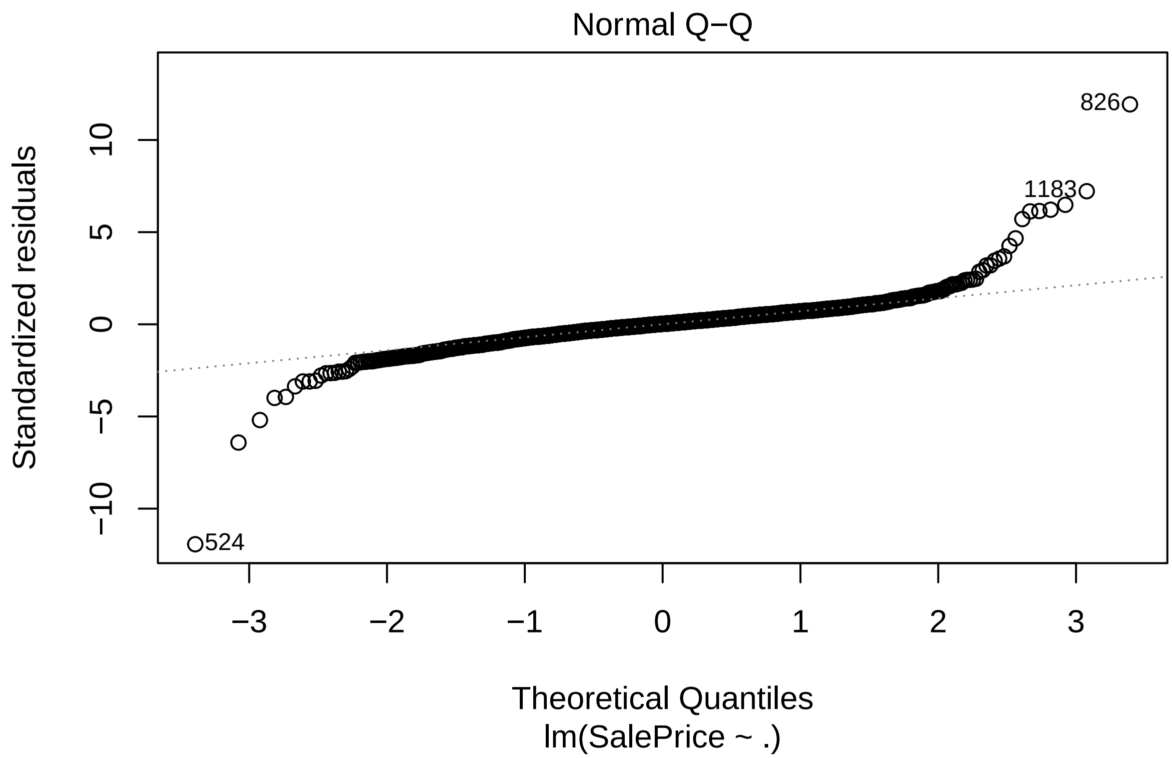
```
##           FenceMnWw          HeatingGasA  NeighborhoodMeadowV
##       -3.005411e+03        -3.196340e+03        -3.214192e+03
##     BsmtFinType1LwQ       Condition2RRNn          HeatingQCTA
##       -3.287536e+03        -3.291909e+03        -3.570916e+03
##        BedroomAbvGr       BsmtExposureMn          HeatingQCGd
##       -3.705098e+03        -3.779971e+03        -3.841663e+03
##  NeighborhoodSawyerW     Exterior2ndStone  NeighborhoodBrkSide
##       -3.858319e+03        -4.109203e+03        -4.567787e+03
##     HouseStyle2Story       BsmtExposureNo             PoolQCFa
##       -5.209518e+03        -5.258552e+03        -5.522601e+03
##          PavedDriveP      ElectricalFuseP       BldgTypeDuplex
##       -5.715624e+03        -5.825336e+03        -6.274401e+03
##          ExterCondFa          LotConfigFR2          HeatingGasW
##       -6.818316e+03        -7.405887e+03        -7.572769e+03
##          ExterCondTA     HouseStyle2.5Unf   NeighborhoodTimber
##       -7.914850e+03        -8.811214e+03        -9.028583e+03
##     Condition2Feedr          ExterQualFa        LandContourLow
##       -9.335109e+03        -9.356369e+03        -9.411838e+03
##    NeighborhoodSWISU  NeighborhoodCollgCr      BsmtFinType2Unf
##       -9.566110e+03        -9.641576e+03        -1.005690e+04
##    Exterior1stStucco          HeatingGrav     Exterior2ndOther
##       -1.014966e+04        -1.036132e+04        -1.040614e+04
##       Condition2Norm   Exterior1stMetalSd   NeighborhoodIDOTRR
##       -1.042745e+04        -1.050093e+04        -1.072342e+04
##   NeighborhoodSawyer   Exterior1stBrkComm          ExterCondGd
##       -1.084710e+04        -1.087669e+04        -1.113206e+04
##      BsmtFinType2Rec      BsmtFinType2BLQ  NeighborhoodGilbert
##       -1.144746e+04        -1.248338e+04        -1.255339e+04
##           BsmtQualFa  NeighborhoodOldTown   Exterior1stWdShing
##       -1.259250e+04        -1.308796e+04        -1.338169e+04
##  NeighborhoodClearCr         KitchenAbvGr    Exterior1stCBlock
##       -1.386768e+04        -1.403196e+04        -1.437834e+04
##           BsmtQualTA   Exterior1stCemntBd     Exterior1stStone
##       -1.487192e+04        -1.507421e+04        -1.507823e+04
##         LotConfigFR3      BsmtFinType2LwQ       BldgTypeTwnhsE
##       -1.516310e+04        -1.528856e+04        -1.564378e+04
##    NeighborhoodNAmes       Condition1RRAe   Exterior1stWd Sdng
##       -1.580953e+04        -1.658622e+04        -1.801754e+04
##           BsmtQualGd   Exterior1stAsphShn    NeighborhoodNWAmes
##       -1.825636e+04        -1.828677e+04        -1.836441e+04
##        BldgTypeTwnhs  NeighborhoodEdwards          ExterQualGd
##       -1.965552e+04        -2.004005e+04        -2.007137e+04
##   Exterior1stVinylSd   Exterior1stHdBoard         KitchenQualFa
##       -2.019854e+04        -2.045566e+04        -2.077965e+04
##  NeighborhoodMitchel   Exterior1stPlywood          ExterQualTA
##       -2.092723e+04        -2.115657e+04        -2.121217e+04
##         HeatingOthW     HouseStyle2.5Fin       Condition2RRAn
##       -2.279578e+04        -2.285243e+04        -2.427136e+04
##        KitchenQualTA        KitchenQualGd       FoundationWood
##       -2.434808e+04        -2.561111e+04        -3.144575e+04
##      UtilitiesNoSeWa      MiscFeatureTenC         FunctionalSev
##       -3.673121e+04        -3.698017e+04        -3.731274e+04
##         LandSlopeSev   Exterior1stImStucc          ElectricalMix
##       -4.377117e+04        -4.426576e+04        -4.998121e+04
```

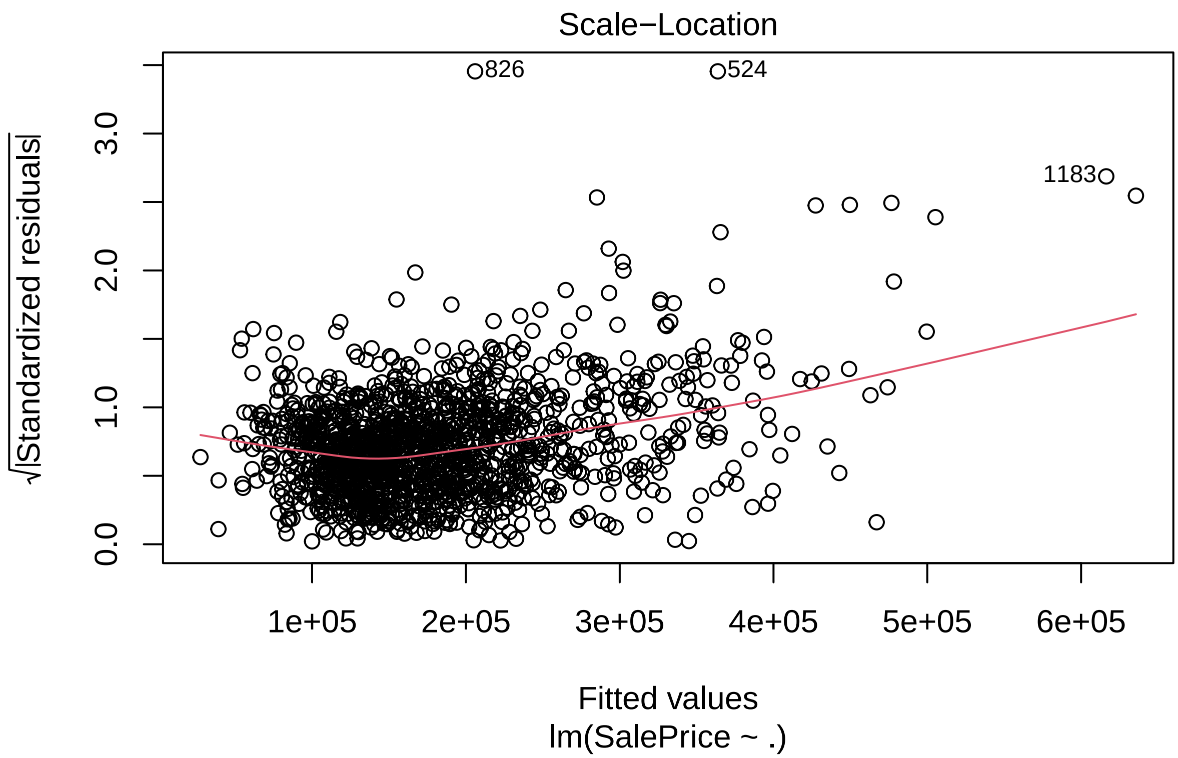
```
##         GarageQualGd         GarageQualTA       Condition2RRAe
##        -1.101734e+05        -1.109383e+05        -1.126567e+05
##         GarageQualFa         GarageQualPo       Condition2PosN
##        -1.193777e+05        -1.335563e+05        -2.394265e+05
##          (Intercept)
##        -5.046280e+05
```

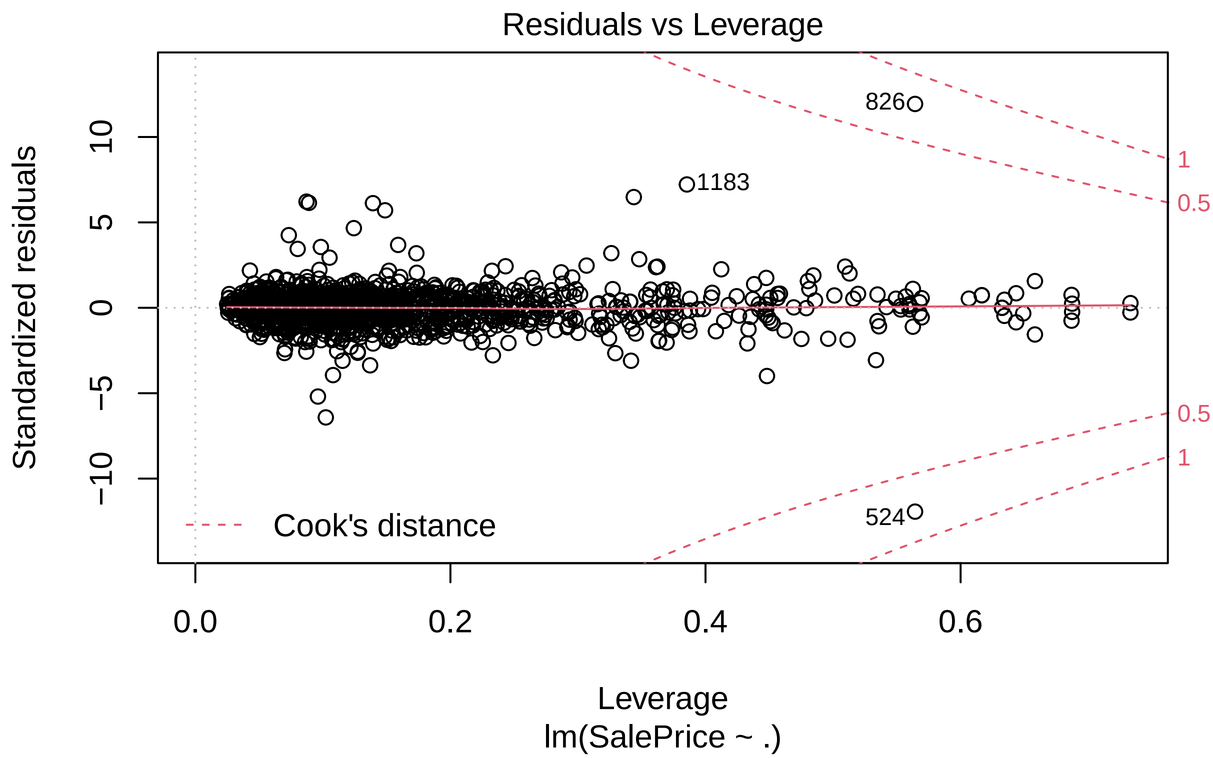
```
plot(ols)
```

```
## Warning: not plotting observations with leverage one:
##   121, 186, 251, 272, 326, 347, 376, 399, 584, 596, 667, 945, 1004, 1012, 1188, 1231, 1271, 1276, 12
```




Normal Q−Q
Standardized residuals
10
5
0
−5
−10
826
1183
524
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(SalePrice ~ .)


Scale−Location
√|Standardized residuals|
0.0
1.0
2.0
3.0
826
524
1183
1e+05
2e+05
3e+05
4e+05
5e+05
6e+05
Fitted values
lm(SalePrice ~ .)



Because of many categorical variables with multiple levels (greater than 2), there are 243 variables when accounting dummy variables. This is not a desirable because it's offsetting one of strengths of simple regression model which showing relationship between explanatory variables and the response variable. For this purpose of analysis, the major focus is accuracy so test MSE is the only measurement used to measure the strength of a model.Furthermore, because normality assumption of residuals is violated as shown in the qqplot of residuals, this OLS doesn't show any relationships between variables.

```
pred.ols <- predict(ols,newdata=data[(nrow(train)+1):nrow(data),])
```

```
## Warning in predict.lm(ols, newdata = data[(nrow(train) + 1):nrow(data), :
## prediction from a rank-deficient fit may be misleading
```

The warning message indicates that there could be potential of multicollinearity which may lead to having rank less than the number of parameters in the model. Multicollinearity should be avoided to prevent the inflation of F-test statistics, and generate a reliable coefficients for each variables in the model. As previously mentioned, only test MSE is considered.

```
test.sale <- read.csv("sample_submission.csv")
test.sale <- test.sale[-692,] # removing the 2152th row of the combined data because of the removal don
nrow(test.sale)
```

```
## [1] 1458
```

```
sqrt(mean((pred.ols - test.sale$SalePrice)^2)) # Test MSE
```

```
## [1] 76994.58
```

The RMSE is about $77,000, which is quiet high in my opinion. Let's suppose there is a house with a true intrinsic value of $500,000. Such value is absolutely correct. With this model, the house could be valued at either $10,000 or $990,000, which shows that such model is not reliable at all. The main reason for this could be that the initial model is overfitting the training data, so when a new data is introduced, it is not adequately accounting them.

Improving OLS with stepwise

Stepwise regression is a method of dropping insignificant variables. There are multiple methods, but in this analysis, forward method is used. Forward method only includes additional variable if and only if adding a variable enhances the model evaluated by F-partial test.

```
library(MASS)

stepfor <- stepAIC(ols,direction="both",trace=F)


sqrt(mean((predict(stepfor,newx=data[(nrow(train)+1):nrow(data),])- test.sale$SalePrice)^2))
```

```
## [1] 77881.16
```

Lasso Regression

Because the data contains many predictors which may possess multicollinearity, lasso regression is tested. Like the ridge regression, it solves multicollinearity by lowering the values of coefficients. Additionally, lasso removes insignificant variables.

```
library("glmnet")
# Finding the best value of lambda

data.lasso <- scale(model.matrix(SalePrice~.,data=data))
y <- (data$SalePrice) #response variable

#finding the best lamdba value with 5-fold cv
set.seed(20220103)
lasso <- cv.glmnet(data.lasso[1:nrow(train),],y[1:nrow(train)],alpha=1,thresh=1e-23, nfolds=5)
 # the value of lambda that gives the lowest mean cross validated error

lasso.mod <- glmnet(data.lasso[1:nrow(train),],y[1:nrow(train)],alpha=1,thresh=1e-23,lambda = min(lasso$

lasso.pred <- predict(lasso.mod,newx=data.lasso[(nrow(train)+1):nrow(data),])

sqrt(mean((lasso.pred-test.sale$SalePrice)^2))
```

```
## [1] 74308.81
```

The main purpose of using lasso regression is to reduce variance by restraining magnitude of variables, even to 0 for some at the cost of increasing bias. The bias might be a structural problem this data contains like having less significant variables either in the dataset or the model. At this stage, the latter might be true because EDA wasn't conducted to select variables. Instead, random selections were done with stepwise then some random selection with coefficient coercion through lasso. Lastly, random forest is tested. Random forest takes subset of variables to split tree. During this process, the hierarchical order of variables is constructed, and having many trees could further reduce the variance by increasing the sample size.

Random Forest

```
library("randomForest")

rf <- randomForest(SalePrice~., data=data[1:nrow(train),], mtry=floor((ncol(data)-1)/3), importance=T)


rf.pred <- predict(rf,newdata=data[(nrow(train)+1):nrow(data),])

sqrt(mean((rf.pred-test.sale$SalePrice)^2))
```

```
## [1] 67590.04
```

RMSE has improved but still reasonably big in my opinion.

```
sort(importance(rf)[,1],decreasing = T)
```

```
##     GrLivArea  Neighborhood   OverallQual   TotalBsmtSF     X1stFlrSF
##   35.18989492   27.32369858   25.28259135   21.16130968   18.50714940
##     X2ndFlrSF    GarageCars    GarageArea    BsmtFinSF1     ExterQual
##   15.57226960   14.71561117   13.69362150   13.27672128   12.41331899
##       LotArea  BsmtFinType1      MSZoning    MSSubClass    CentralAir
##   10.74219970    9.80039521    9.66787797    9.65938018    9.27986099
##    GarageType     YearBuilt  GarageFinish   KitchenQual    Fireplaces
##    9.07835066    9.07027290    9.00807052    8.41134001    8.23121898
##     BsmtUnfSF      BsmtQual      FullBath    HouseStyle  YearRemodAdd
##    8.14238519    7.68381389    7.49363562    7.31340961    7.04119104
##      HalfBath   GarageYrBlt  BedroomAbvGr  TotRmsAbvGrd  KitchenAbvGr
##    7.00030543    6.94736769    6.83353422    6.72965929    6.33039680
##  BsmtFullBath    MasVnrArea      BldgType   Exterior1st   OverallCond
##    6.30076088    5.93720978    5.88717867    5.76078856    5.60633945
##    WoodDeckSF   OpenPorchSF     HeatingQC   LotFrontage  BsmtExposure
##    5.37254147    5.09506739    4.70380955    4.34117448    4.10587573
##    Foundation         Fence   Exterior2nd         Alley    MasVnrType
##    4.08861051    3.98783430    3.98059936    3.53228429    3.09467072
##    BsmtFinSF2      BsmtCond    Functional     LandSlope      LotShape
##    2.83821055    2.80458995    2.72065147    2.63078557    2.23960542
##        PoolQC      SaleType   FireplaceQu    PavedDrive    Condition1
##    2.16745324    2.14305566    2.05575191    2.05399306    2.02570704
## SaleCondition     RoofStyle   ScreenPorch   LandContour EnclosedPorch
##    1.87941130    1.74442610    1.57582911    1.35223171    1.30537947
##  BsmtHalfBath        Street   MiscFeature       MiscVal    Electrical
```

```
##    1.10847472    1.04634466    1.00100150    0.91020006    0.59559615
##        YrSold    X3SsnPorch        MoSold     Utilities     ExterCond
##    0.38924706    0.31081738    0.21070573    0.00000000   -0.03506364
##       Heating    GarageQual     LotConfig    Condition2      PoolArea
##   -0.13098427   -0.58571498   -0.77574756   -1.14792979   -1.26080263
##      RoofMatl   BsmtFinType2    GarageCond  LowQualFinSF
##   -1.35110516   -1.44558093   -1.62260927   -1.66171309
```

The numbers above show the order of important variables sorted by decrease in training MSE when not included. The negative values mean that excluding them will increase training MSE. One can genuinely inference that choosing selecting predictors based on this is viable since EDA might be challenging when the dataset has around 80 predictors.

```r
important.var <- names(sort(importance(rf)[,1],decreasing = T)>5) # names of variables that have greate
```

Tests with only important Var

```r
data2 <- data.frame(cbind(data$SalePrice, data[,important.var]))
ols2 <- lm(data.SalePrice~.,data=data2[1:nrow(train),])

ols2.pred <- predict(ols2,newdata=data2[(nrow(train)+1):nrow(data),])
```

```
## Warning in predict.lm(ols2, newdata = data2[(nrow(train) + 1):nrow(data), :
## prediction from a rank-deficient fit may be misleading
```

```r
sqrt(mean((ols2.pred-test.sale$SalePrice)^2)) # lower than the original but high
```

```
## [1] 76994.58
```

```r
data2.lasso <- scale(model.matrix(data.SalePrice~.,data=data2))
y2 <- data2$data.SalePrice

set.seed(20220103)
lasso2 <- cv.glmnet(data2.lasso[1:nrow(train),],y[1:nrow(train)],alpha=1,thresh=1e-23, nfolds=5)
```

```
## Warning: from glmnet Fortran code (error code -89); Convergence for 89th lambda
## value not reached after maxit=100000 iterations; solutions for larger lambdas
## returned
```

```
## Warning: from glmnet Fortran code (error code -93); Convergence for 93th lambda
## value not reached after maxit=100000 iterations; solutions for larger lambdas
## returned
```

```
## Warning: from glmnet Fortran code (error code -88); Convergence for 88th lambda
## value not reached after maxit=100000 iterations; solutions for larger lambdas
## returned
```

```
## Warning: from glmnet Fortran code (error code -91); Convergence for 91th lambda
## value not reached after maxit=100000 iterations; solutions for larger lambdas
## returned

## Warning: from glmnet Fortran code (error code -91); Convergence for 91th lambda
## value not reached after maxit=100000 iterations; solutions for larger lambdas
## returned
```

```
 # the value of lambda that gives the lowest mean cross validated error

lasso.mod2 <- glmnet(data2.lasso[1:nrow(train),],y2[1:nrow(train)],alpha=1,thresh=1e-23,lambda = min(la

lasso.pred2 <- predict(lasso.mod2,newx=data2.lasso[(nrow(train)+1):nrow(data),])

sqrt(mean((lasso.pred2-test.sale$SalePrice)^2)) # better than before
```

```
## [1] 74136.38
```

```
#rf
rf2 <- randomForest(data.SalePrice~., data=data2[1:nrow(train),], mtry=floor(ncol(data2)/3), importance


rf.pred2 <- predict(rf2,newdata=data2[(nrow(train)+1):nrow(data),])

sqrt(mean((rf.pred2-test.sale$SalePrice)^2))
```

```
## [1] 67380.42
```

EDA

```
library("corrplot")

train.f <- data[1:nrow(train),]

train.f.num <- Filter(is.numeric,train.f)

#abs(cor(train.f.num))>=0.85 & abs(cor(train.f.num))<1
```

GarageArea and GarageCars have higher correlation than 0.85 so likely to cause multicollinearity issue. Hence, GarageCars is removed because GarageArea is more self explanatory.

Dropping GarageCars

```
data <- dplyr::select(data,-GarageCars)
train.f.num <- Filter(is.numeric,data)
```

Nonlinearity

Spear's Rank Correlation

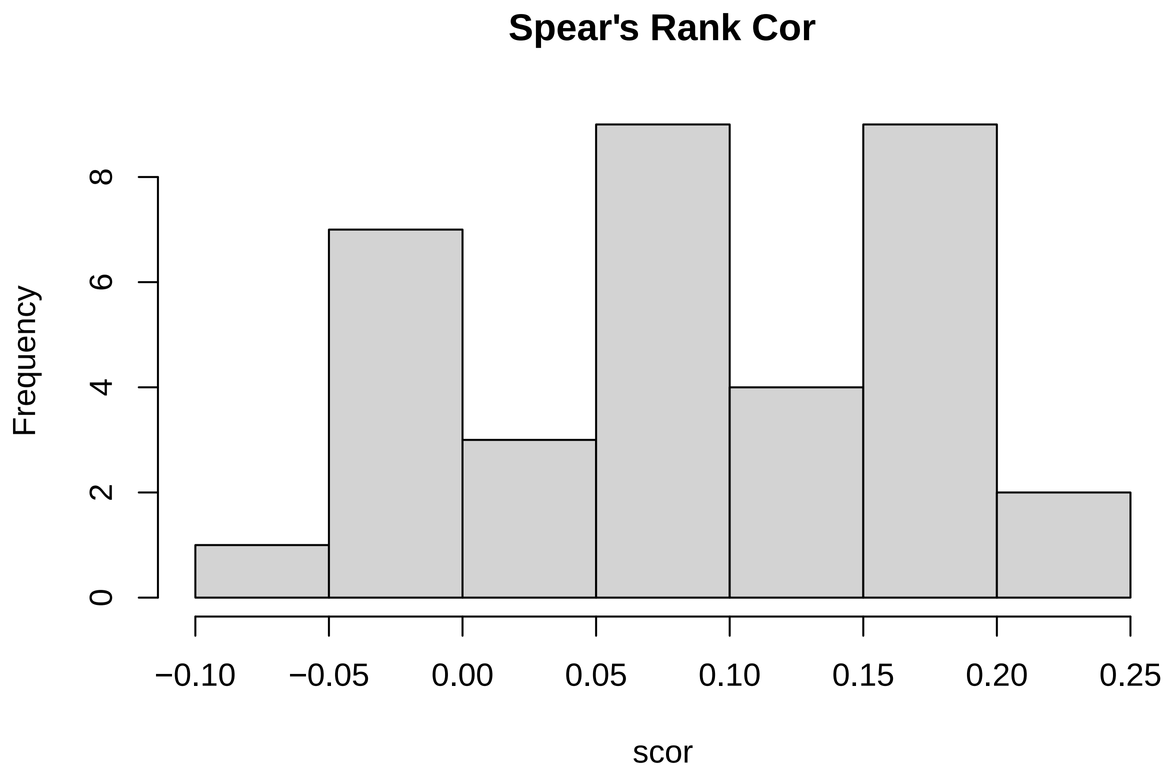
```
# Spear's Rank Correlation

#which(colnames(train.f.num)=="SalePrice") 36 is the column number of SalePrice

num.col.n <- colnames(train.f.num)[1:35]

scor <- matrix(NA,nrow=35)


for( i in 1:35){
  scor[i] <- cor(rank(train.f.num$SalePrice),rank(train.f.num[,i]))
}

rownames(scor) <- num.col.n
colnames(scor) <- "SalePrice"

scor[rank(scor)]
```

```
##  [1]  1.734979e-01  1.837841e-01  1.654145e-01  7.978188e-03  1.406142e-01
##  [6]  5.099707e-02 -2.820341e-05 -3.842595e-02  5.544350e-02  2.229113e-01
## [11]  1.201937e-02  1.172989e-01  1.782238e-01  9.699939e-02 -2.348369e-02
## [16]  5.202087e-02  9.982399e-02 -3.484159e-02 -8.986660e-02 -1.799993e-02
## [21]  2.227337e-01  1.265444e-01 -1.556818e-02  9.827310e-02  1.618092e-01
## [26]  2.102091e-02  1.697221e-01  7.888850e-02 -2.945166e-03  1.676476e-01
## [31]  1.211418e-01  6.421058e-02  1.706892e-01  1.627603e-01  8.539816e-02
```

```
rownames(scor)[rank(scor)]
```

```
##  [1] "YearRemodAdd"  "TotRmsAbvGrd"  "Fireplaces"    "YrSold"
##  [5] "LotArea"       "X3SsnPorch"    "ScreenPorch"   "KitchenAbvGr"
##  [9] "BsmtFullBath"  "OverallQual"   "LowQualFinSF"  "OpenPorchSF"
## [13] "GarageArea"    "HalfBath"      "BsmtFinSF2"    "MoSold"
## [17] "X2ndFlrSF"     "OverallCond"   "EnclosedPorch" "BsmtHalfBath"
## [21] "GrLivArea"     "LotFrontage"   "MiscVal"       "WoodDeckSF"
## [25] "GarageYrBlt"   "PoolArea"      "FullBath"      "BedroomAbvGr"
## [29] "MSSubClass"    "TotalBsmtSF"   "MasVnrArea"    "BsmtUnfSF"
## [33] "YearBuilt"     "X1stFlrSF"     "BsmtFinSF1"
```

```
hist(scor, main= "Spear's Rank Cor")
```



The histogram shows Spear's rank correlation between SalePrice and each numeric variables to identify nonlinear relatinoship between them. Since these values lie between -0.1 and 0.25, it's hard to state that there's nonlinear relationship to introduce higher order terms.To veritfy this result, Kendall's Tau correlation is also checked.

Kendall's Tau correlation

```
Tau <- function(x,y){
  n <- length(x)
  mat <- cbind(x,y)
  mat <- mat[order(mat[,1]),]
  concord <- 0
  for(i in 1:(n-1)){
    for(j in (i+1):n)
    {
      tmp = (x[i]-x[j])*(y[i]-y[j])
      concord = concord + 1*(tmp>0) + 1/2*(tmp==0)
    }
  }
  2*concord/(n*(n-1)/2) - 1
}

tcor <- matrix(NA,nrow=35)
```

```r
for (i in 1:35){
  tcor[i] <- Tau(train.f.num$SalePrice, train.f.num[,i])
}

tcor
```

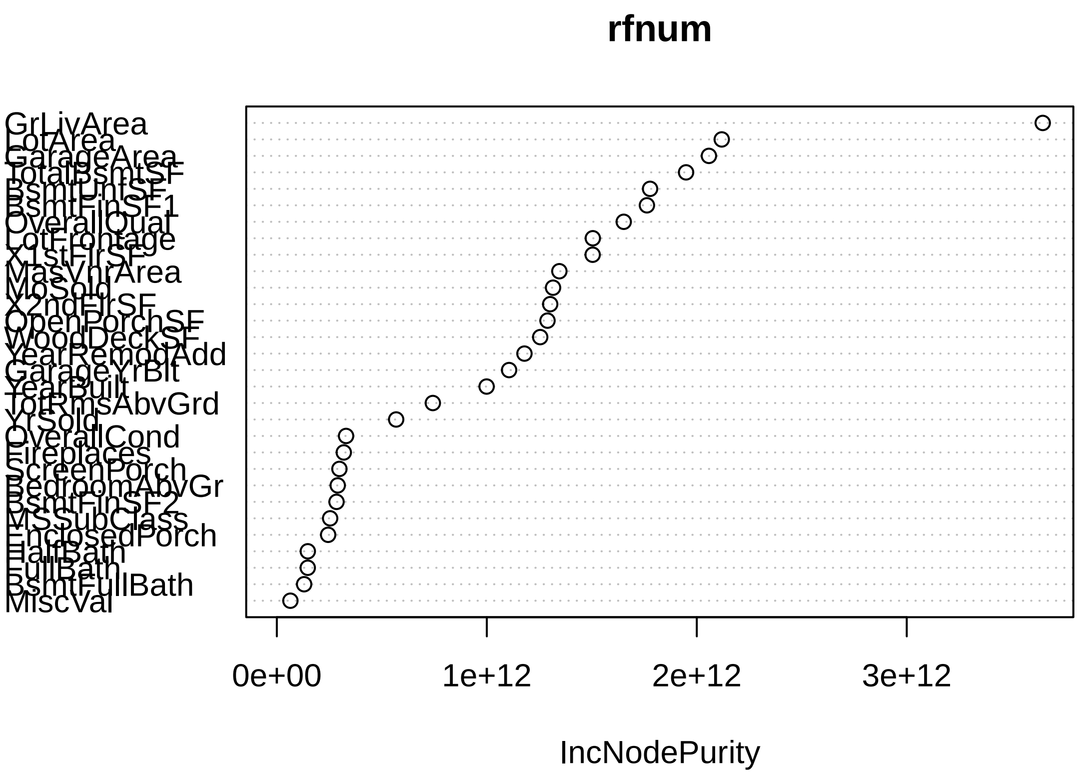
```
##               [,1]
##  [1,] -0.003382361
##  [2,]  0.082212635
##  [3,]  0.090320198
##  [4,]  0.154361601
##  [5,] -0.020182321
##  [6,]  0.114965026
##  [7,]  0.114758020
##  [8,]  0.069096030
##  [9,]  0.055729889
## [10,] -0.008235385
## [11,]  0.040938668
## [12,]  0.113857388
## [13,]  0.108903798
## [14,]  0.059800235
## [15,]  0.001506848
## [16,]  0.150673312
## [17,]  0.029754438
## [18,] -0.004618526
## [19,]  0.093551474
## [20,]  0.050682076
## [21,]  0.045494693
## [22,] -0.008693807
## [23,]  0.116112844
## [24,]  0.092683268
## [25,]  0.107210620
## [26,]  0.121981868
## [27,]  0.057314511
## [28,]  0.072032187
## [29,] -0.035203105
## [30,]  0.006175188
## [31,] -0.000043704
## [32,]  0.001513192
## [33,] -0.003113793
## [34,]  0.032572406
## [35,]  0.004930798
```

Kendall's Tau is also not showing significant nonlinearity between SalePrice and numeric variables. Hence, it's hard to justify using higher order terms.

Interaction between continuous variables

To identify if interactions between continuous variables exist, conditional plot is used. Since there are 35 numeric predictors, there are 595 pairs need to be evaluated for possible interactions which is extremely time consuming. To focus on few, randomforest is used with only numeric predictors to identify the most important predictors. Then, interactions will be evaluated.

```r
set.seed(20220105)
rfnum <- randomForest(SalePrice ~., data=train.f.num, mtry=floor((ncol(data)-1)/3))
varImpPlot(rfnum)
```



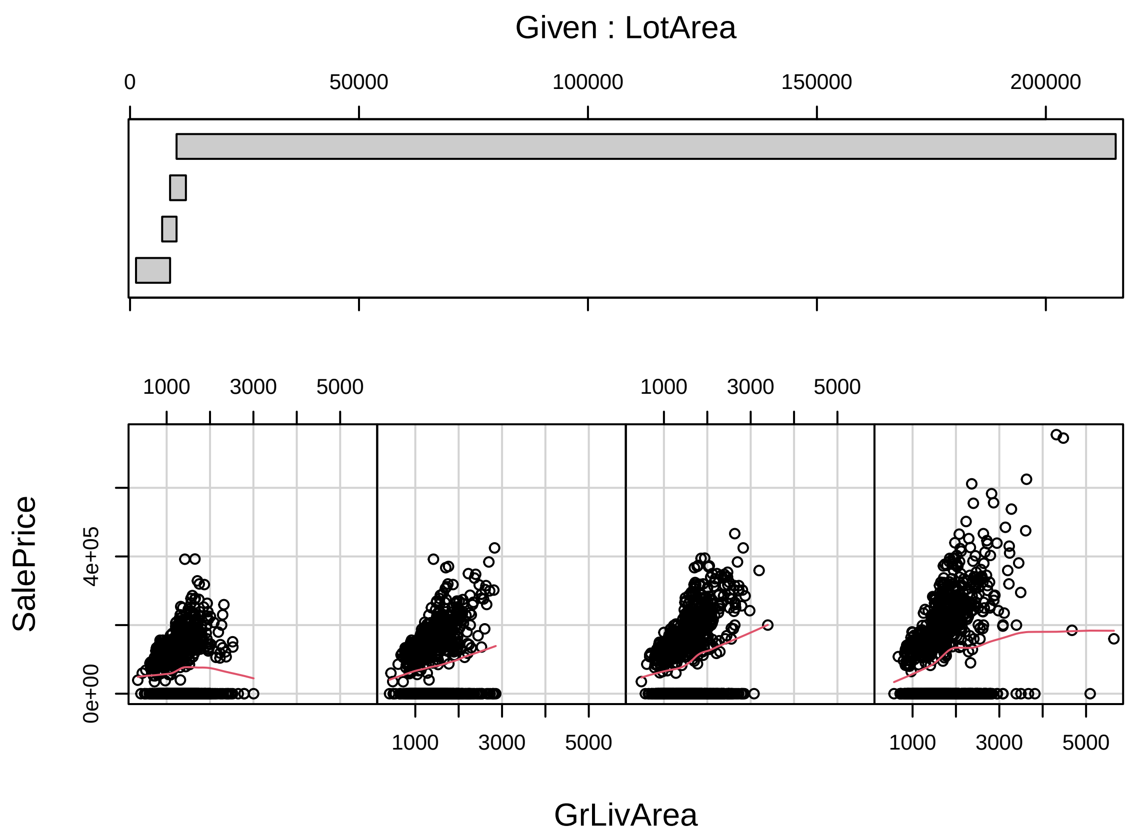

```r
imp5 <- names(sort(importance(rfnum)[,1],decreasing = T))[1:5]
```

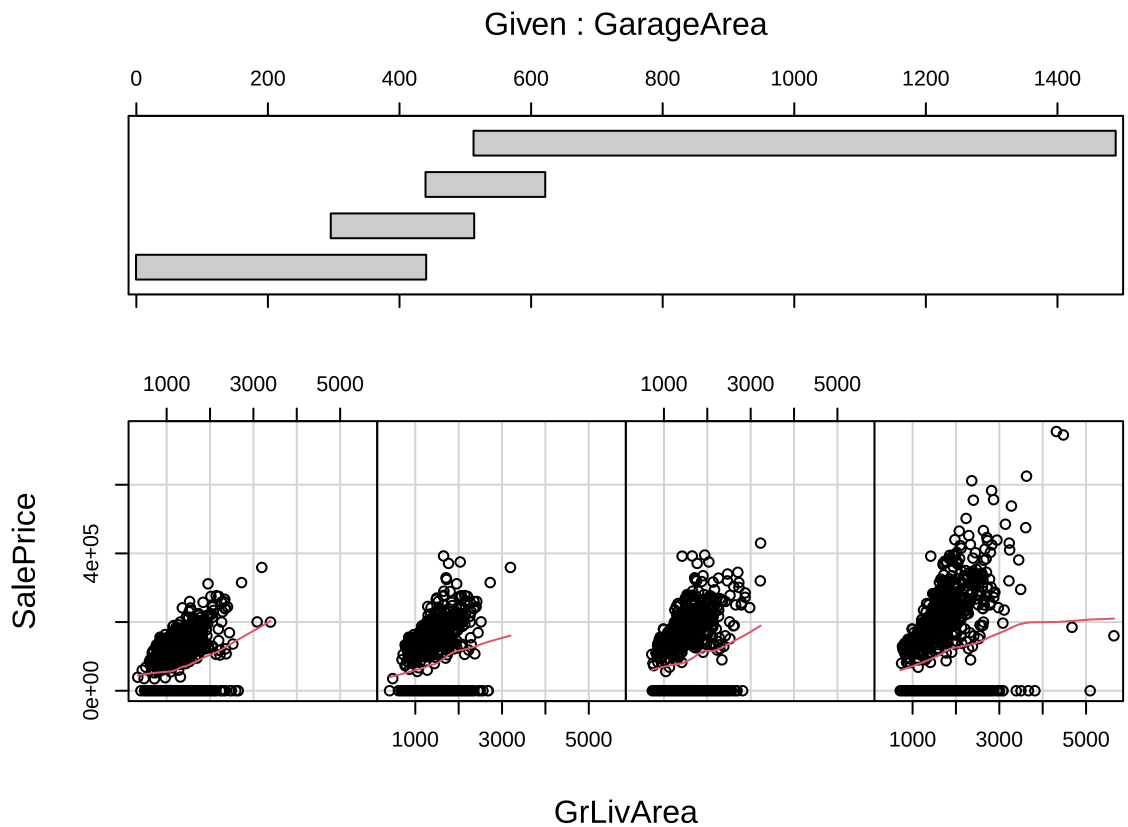
The most important 5 predictors will be evaluated.

```r
library("lattice")

colnames(train.f.num)
```

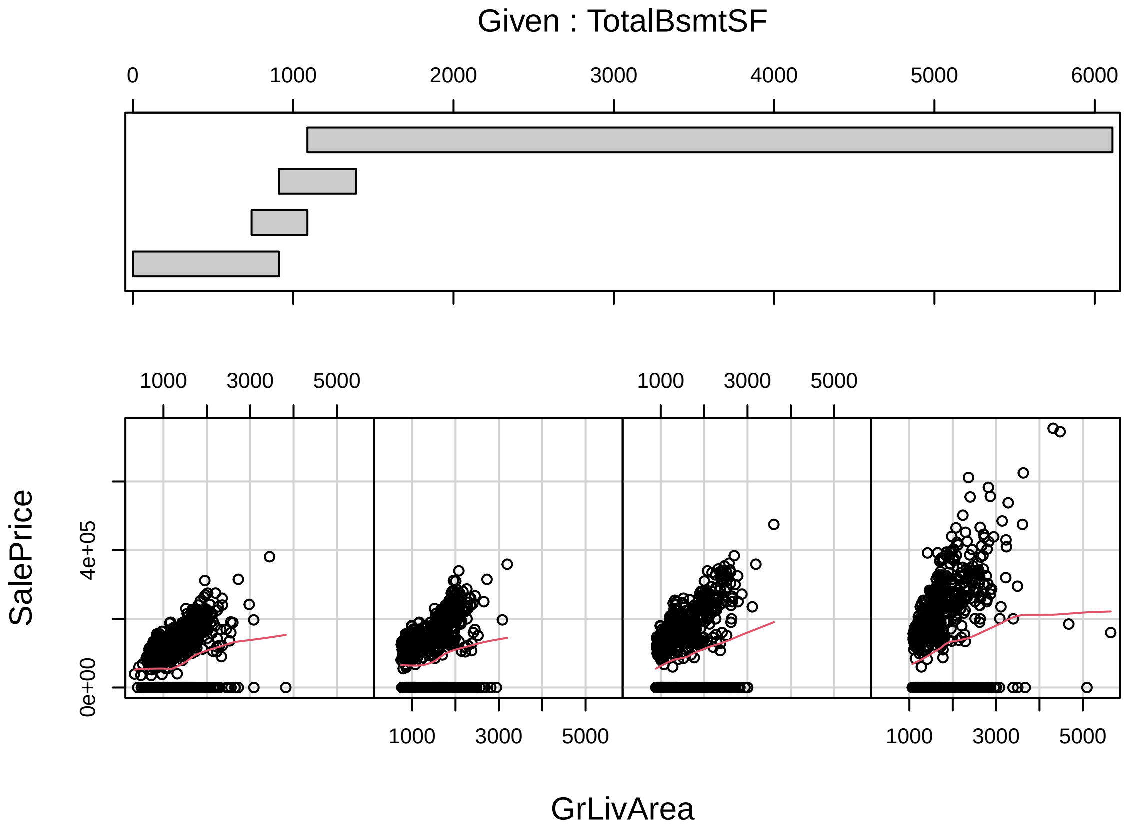
```
##  [1] "MSSubClass"    "LotFrontage"   "LotArea"       "OverallQual"
##  [5] "OverallCond"   "YearBuilt"     "YearRemodAdd"  "MasVnrArea"
##  [9] "BsmtFinSF1"    "BsmtFinSF2"    "BsmtUnfSF"     "TotalBsmtSF"
## [13] "X1stFlrSF"     "X2ndFlrSF"     "LowQualFinSF"  "GrLivArea"
## [17] "BsmtFullBath"  "BsmtHalfBath"  "FullBath"      "HalfBath"
## [21] "BedroomAbvGr"  "KitchenAbvGr"  "TotRmsAbvGrd"  "Fireplaces"
## [25] "GarageYrBlt"   "GarageArea"    "WoodDeckSF"    "OpenPorchSF"
## [29] "EnclosedPorch" "X3SsnPorch"    "ScreenPorch"   "PoolArea"
## [33] "MiscVal"       "MoSold"        "YrSold"        "SalePrice"
```

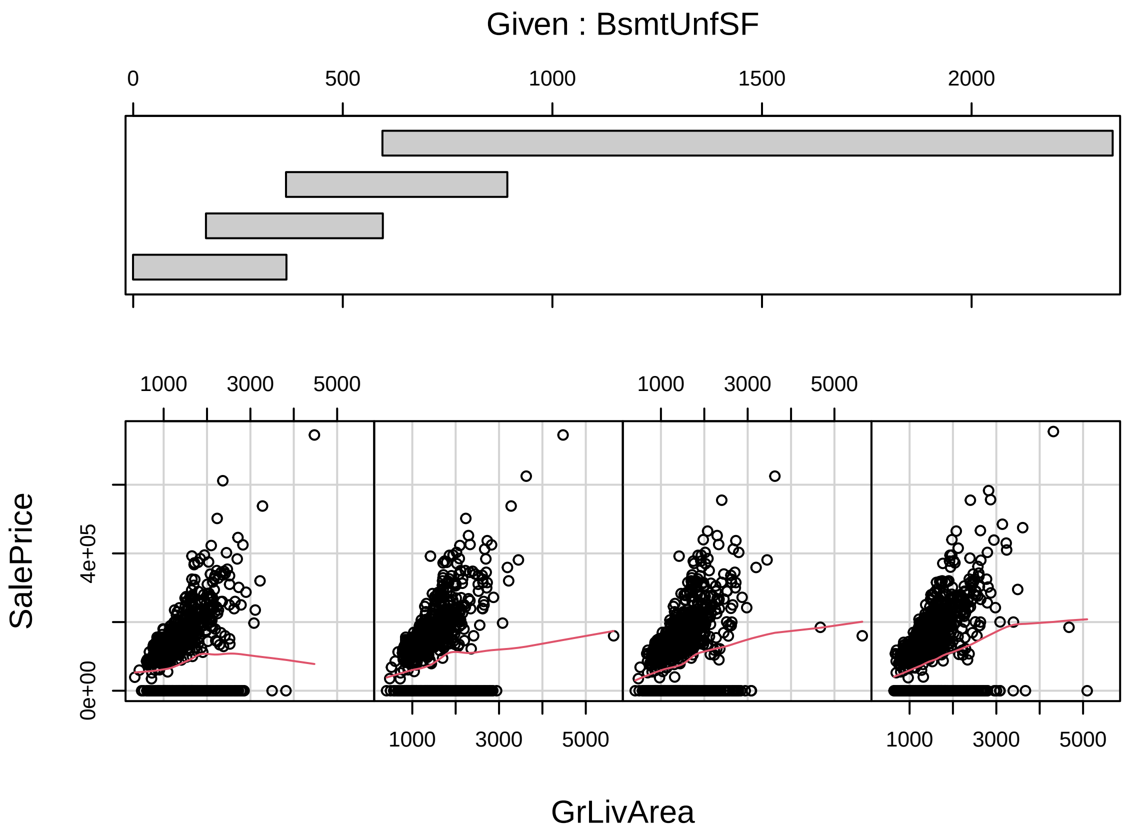
```r
for(i in 1:5){
  if( i <5){
  for (j in 1:5){
    if(j>i){
   coplot(as.formula(paste(paste(paste("SalePrice~",imp5[i]),"|"),imp5[j])),
       number = 4, rows = 1,
       panel = panel.smooth, data=train.f.num)
  }
  }
  }
}
```

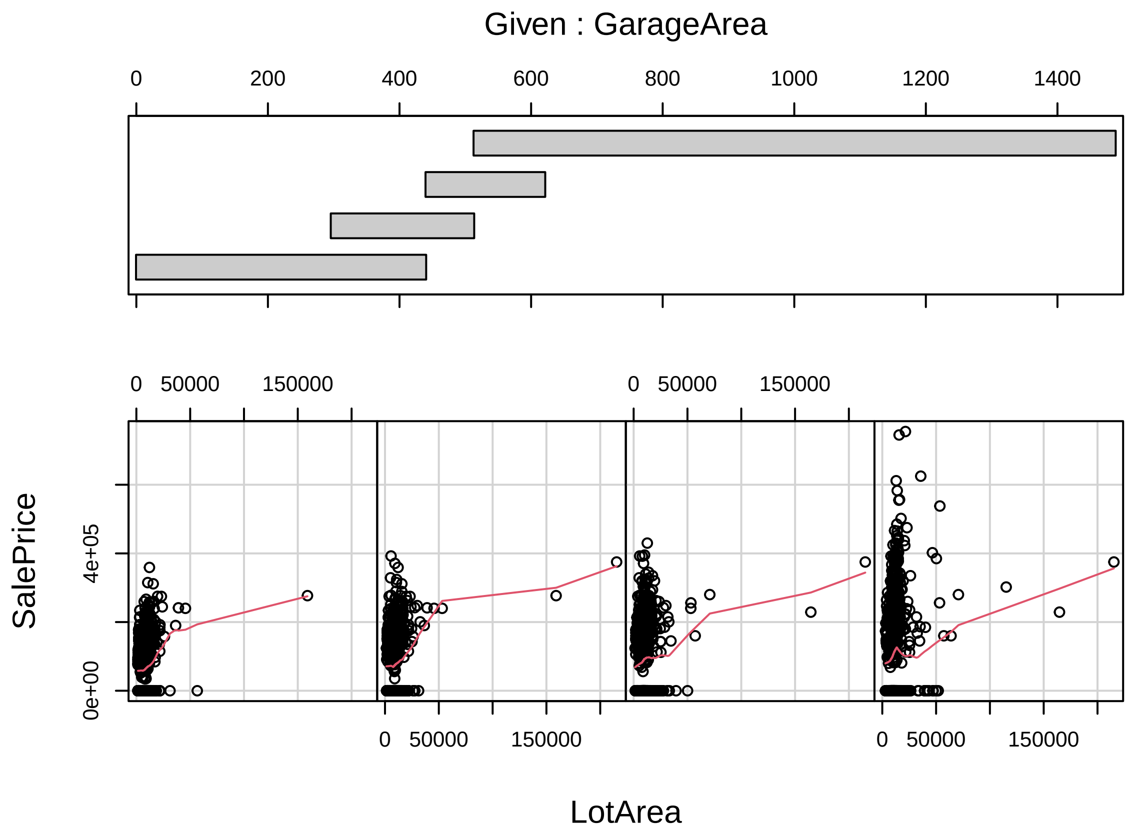


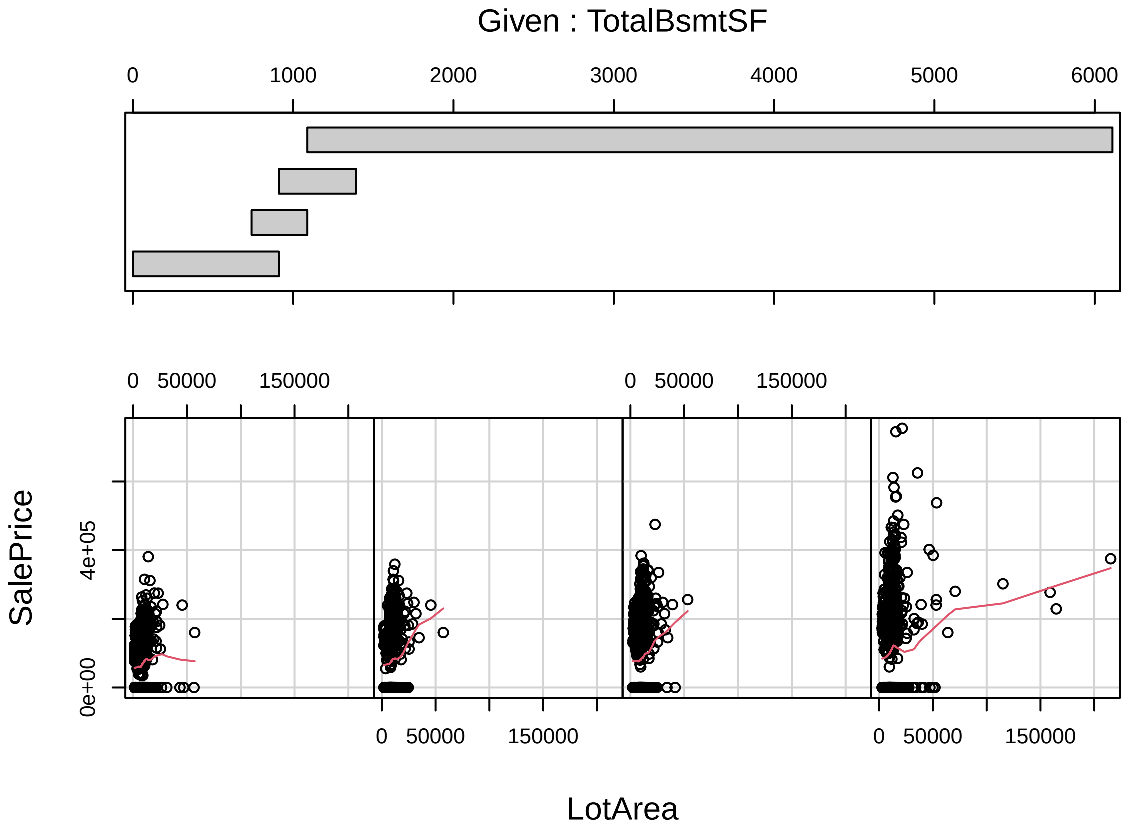


Given : GarageArea
0
200
400
600
800
1000
1200
1400
1000
3000
5000
SalePrice
4e+05
0e+00
GrLivArea

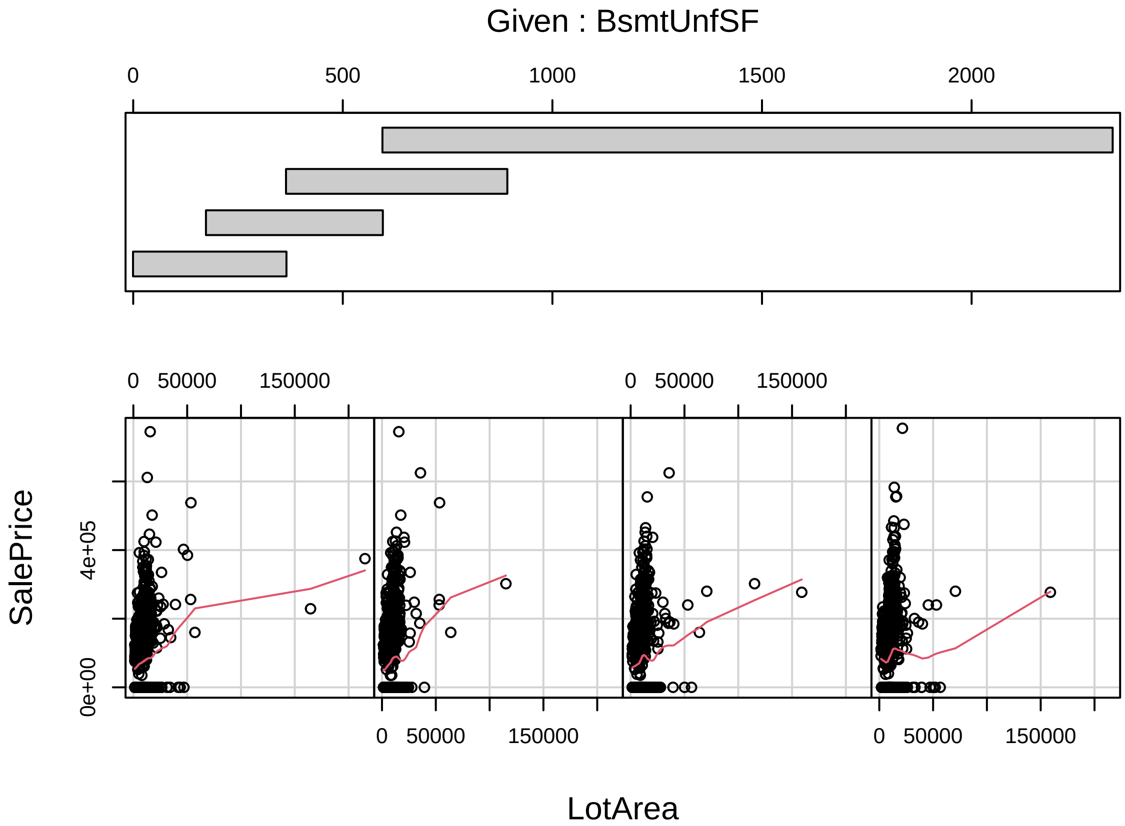

Given : TotalBsmtSF
0
1000
2000
3000
4000
5000
6000
1000
3000
5000
SalePrice
0e+00
4e+05
GrLivArea

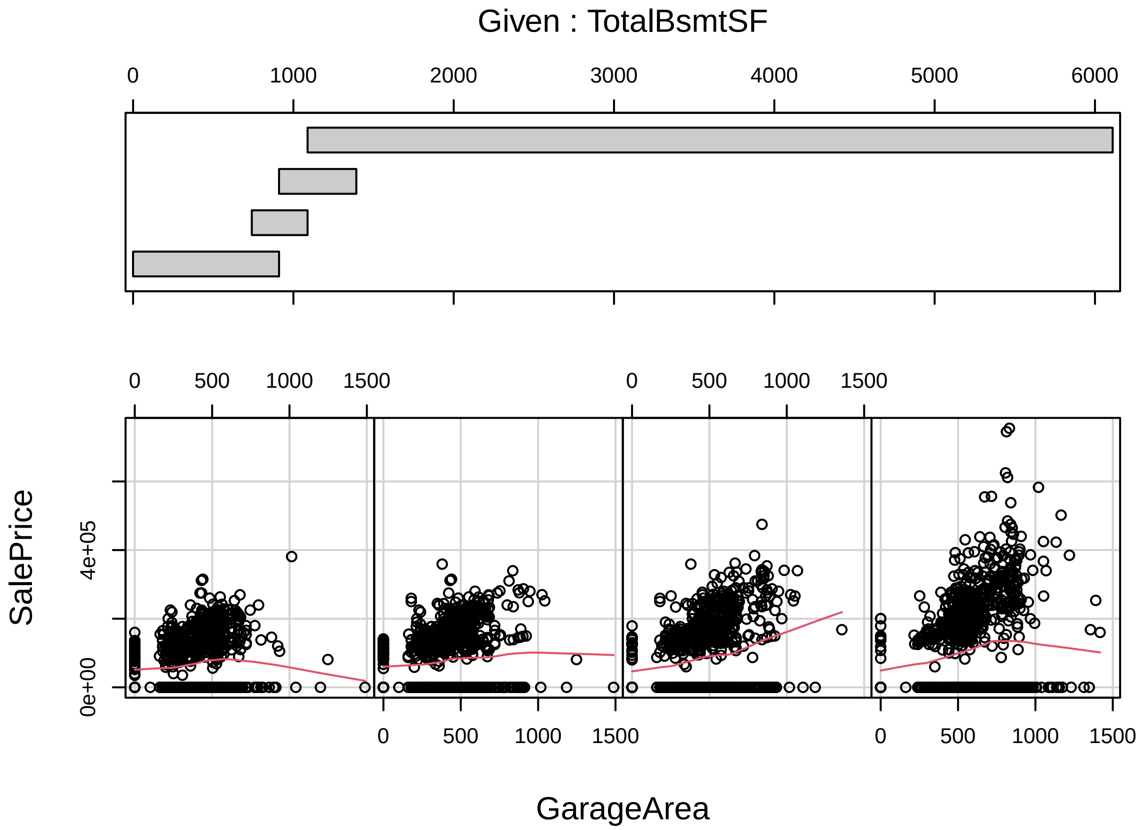

Given : BsmtUnfSF
0
500
1000
1500
2000
1000
3000
5000
SalePrice
0e+00
4e+05
GrLivArea

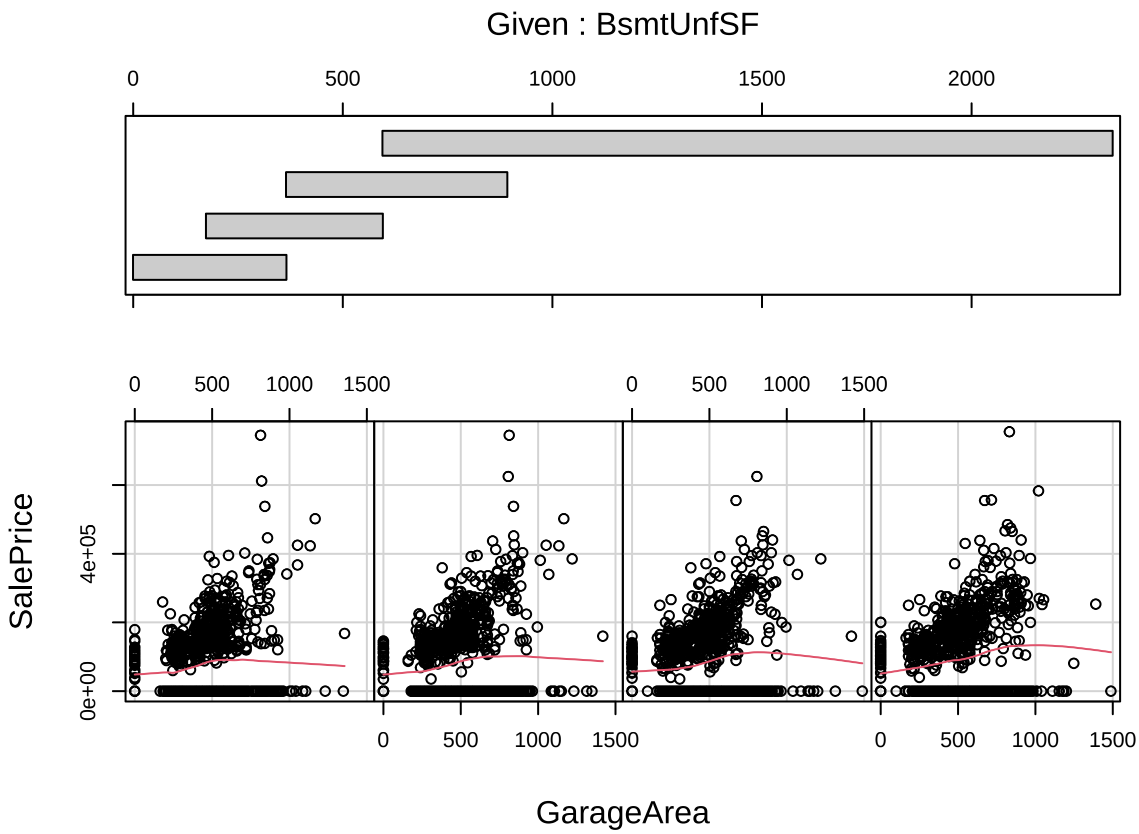

Given : GarageArea
0
200
400
600
800
1000
1200
1400
0
50000
150000
SalePrice
0e+00
4e+05
LotArea

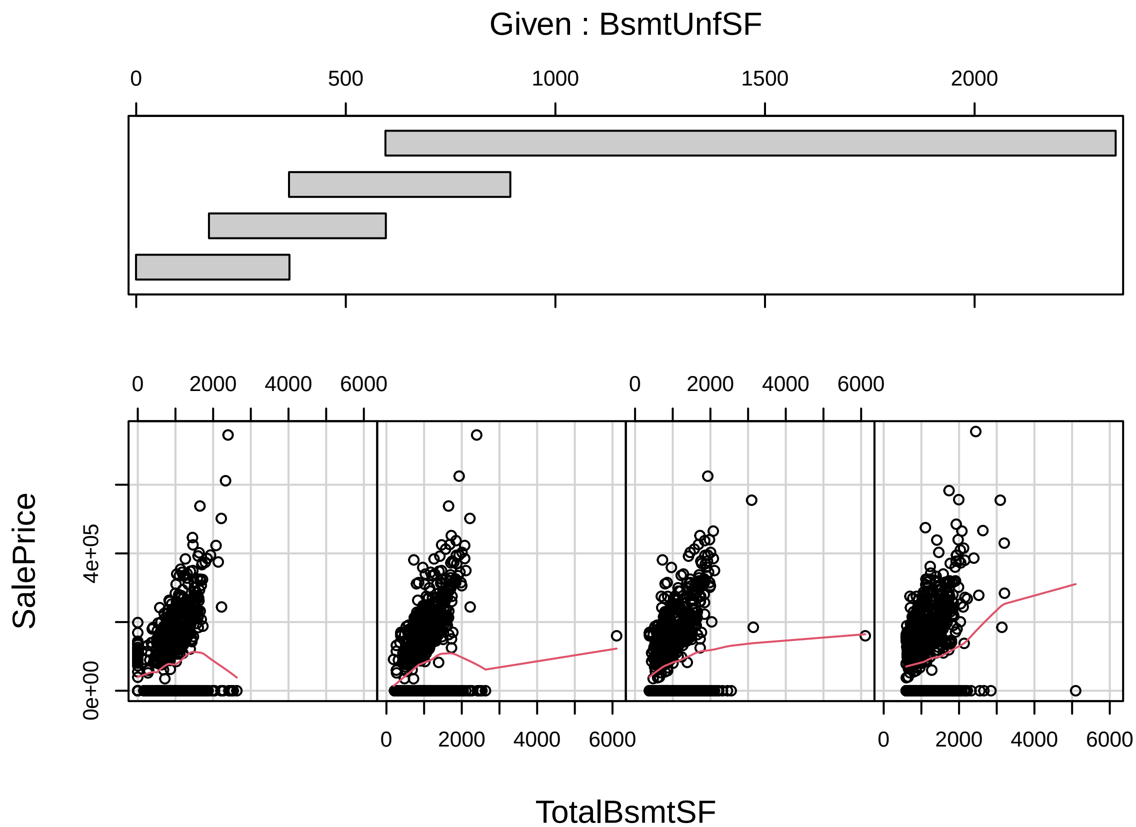


Given : TotalBsmtSF
0
1000
2000
3000
4000
5000
6000
0 50000 150000
0 50000 150000
SalePrice
4e+05
0e+00
0 50000 150000
0 50000 150000
LotArea


Given : BsmtUnfSF
0
500
1000
1500
2000
SalePrice
0e+00
4e+05
0
50000
150000
LotArea


Given : TotalBsmtSF
0
1000
2000
3000
4000
5000
6000
0
500
1000
1500
SalePrice
4e+05
0e+00
GarageArea


Given : BsmtUnfSF
0
500
1000
1500
2000
SalePrice
0e+00
4e+05
GarageArea
0
500
1000
1500



On the plot, interaction is shown if the trend of subdivided data by another predictor is signified to either direction when the value of such predictor is increased.

GrLivArea & Lot Area The relationship hasn't signified. Thus, no interaction.

GrLivArea & TotalBsmtSF The relationship hasn't signified. Thus, no interaction.

GrLivArea & BsmtFinSF1 The relationship hasn't signified. Thus, no interaction.

GrLivArea & BsmtUnfSF The relationship hasn't signified. Thus, no interaction.

LotArea & TotalBsmtSF The relationship hasn't signified. Thus, no interaction.

LotArea & BsmtUnfSF The relationship hasn't signified. Thus, no interaction.

TotalBsmtSF & BsmftFinSF1 The relationship has. Thus, interaction exists.

TotalBsmtSF & BsmtUnfSF The relationship has. Thus, interaction exists.

BsmtFinSF1 & BsmtUnfSF The relationship hasn't signified. Thus, no interaction.

Testing interaction

```
var.int <- paste(colnames(train.f.num[1:35]),"+")

ols.int <- lm(as.formula(c("SalePrice ~",var.int,"TotalBsmtSF * BsmtUnfSF +","TotalBsmtSF * BsmtUnfSF")))
```

```
## Warning: Using formula(x) is deprecated when x is a character vector of length > 1.
##   Consider formula(paste(x, collapse = " ")) instead.
```

```r
sqrt(mean((predict(ols.int,newdata=data[(nrow(train)+1):nrow(data),]) - test.sale$SalePrice)^2))
```

```
## Warning in predict.lm(ols.int, newdata = data[(nrow(train) + 1):nrow(data), :
## prediction from a rank-deficient fit may be misleading
```

```
## [1] 98022.78
```

```r
sqrt(mean((predict(lm(SalePrice~.,data=train.f.num),newdata=data[(nrow(train)+1):nrow(data),]) - test.sa
```

```
## Warning in predict.lm(lm(SalePrice ~ ., data = train.f.num), newdata =
## data[(nrow(train) + : prediction from a rank-deficient fit may be misleading
```

```
## [1] 97997.6
```

```r
set.seed(20220105)
rf.interact <- randomForest(as.formula(c("SalePrice ~",var.int,"TotalBsmtSF * BsmtUnfSF +","TotalBsmtSF
```

```
## Warning: Using formula(x) is deprecated when x is a character vector of length > 1.
##   Consider formula(paste(x, collapse = " ")) instead.
```

```
## Warning: Using formula(x) is deprecated when x is a character vector of length > 1.
##   Consider formula(paste(x, collapse = " ")) instead.
```

```r
rf.pred.int <- predict(rf.interact,newdata=data[(nrow(train)+1):nrow(data),])

sqrt(mean((rf.pred.int-test.sale$SalePrice)^2))
```

```
## [1] 142423.8
```

Interactions didn't improve, in fact worsened, test RMSE with multiple regression.

Back to OLS

```r
sim.ols <- lm(SalePrice~GrLivArea, data=data[1:nrow(train),])
sim.ols.pred <- predict(sim.ols,newdata=data[(nrow(train)+1):nrow(data),])

sqrt(mean((sim.ols.pred - test.sale$SalePrice)^2))
```

```
## [1] 44799.66
```

```r
varpaste <- paste(paste(imp5[1:4],"+"))

mult.ols <- lm(formula(paste(c("SalePrice~",varpaste,imp5[5])), collapse=" "),data=data[1:nrow(train),]
```

```
## Warning: Using formula(x) is deprecated when x is a character vector of length > 1.
##   Consider formula(paste(x, collapse = " ")) instead.
```

```
mult.ols.pred <- predict(mult.ols,newdata=data[(nrow(train)+1):nrow(data),])
sqrt(mean((mult.ols.pred - test.sale$SalePrice)^2))
```

```
## [1] 57395.54
```

From all of the models tested, the best performing model is a simple linear regression model with just GrLivArea. Intuitively, this makes sense because the bigger house would have higher costs, so the price should be more expensive. All of these other predictors in the data might contain multicollinearity. For example, bigger houses should have more bedrooms, garage size, and the list goes on. Going back to important variables generated by randomforest, each variable will be added only and if only adding them improve test RMSE.

OLS Automation

```
important.var.nogrlivarea <- important.var[-which(important.var == "GrLivArea")]
important.var.nogrlivarea <- important.var.nogrlivarea[-which(important.var.nogrlivarea == "GarageCars")

rmsepara <-sqrt(mean((sim.ols.pred - test.sale$SalePrice)^2))
modelpara <- unlist(strsplit(toString(sim.ols$call),","))[2]







rmsepara <-sqrt(mean((sim.ols.pred - test.sale$SalePrice)^2))
modelpara <- unlist(strsplit(toString(sim.ols$call),","))[2]





for (i in important.var.nogrlivarea){
  formulapara <- (c(modelpara,paste("+",i)))
  formulapara2 <- as.formula(paste(formulapara,collapse = ""))
  ols.mod <- lm(formulapara2, data=data[1:nrow(train),])
  testpred <- predict(ols.mod,newdata=data[(nrow(train)+1):nrow(data),])
  test.rmse <- sqrt(mean((testpred-test.sale$SalePrice)^2))
  print(test.rmse)

  if ( test.rmse < rmsepara){
    rmsepara <- test.rmse
    modelpara <- formulapara2
  }
}
```

```
## [1] 64462
## [1] 63341.63
## [1] 52907.7
## [1] 50175.23
## [1] 49424.49
```

```
## [1] 54004.8
## [1] 49803.88
## [1] 60030.17
## [1] 43909.44
## [1] 52572.25
## [1] 47910.22
## [1] 44593.34
## [1] 47502.2
## [1] 49692.01
## [1] 57540.43
## [1] 51834.67
## [1] 58272
## [1] 46428.92
## [1] 43993.14
## [1] 61497.46
## [1] 46329.81
## [1] 51988.88
## [1] 54129.18
## [1] 43894.67
## [1] 55416.34
## [1] 50825.59
## [1] 45578.87
## [1] 46593.75
## [1] 48325.02
## [1] 48288.6
## [1] 47318.84
## [1] 51405.33
## [1] 44079.35
## [1] 46168.92
## [1] 45051.74
## [1] 50978.26
## [1] 43763.64
## [1] 50802.57
## [1] 54335.51
## [1] 46160.01
## [1] 51583.77
## [1] 45230.7
## [1] 49827.15
## [1] 43786.54
## [1] 45710
## [1] 45721.97
## [1] 43673.49
## [1] 45454.46
## [1] 45850.37
## [1] 49051.99
## [1] 45525.31
## [1] 48190.31
## [1] 47162.62
## [1] 49228.1
## [1] 45454.2
## [1] 43756.26
## [1] 46564.28
## [1] 45616.62
## [1] 43711.77
```

```
## [1] 43757.92
## [1] 43917.46
## [1] 43420.5
## [1] 45062.61
## [1] 43407.02
## [1] 43393.98
## [1] 43327
## [1] 43321.43
## [1] 44495.42
## [1] 43716.56
## [1] 45606.33
## [1] 44264.39
## [1] 44285.77
## [1] 43621.72
## [1] 45907.54
## [1] 43868.32
## [1] 44908.33
## [1] 44899.74
```

```
modelpara
```

```
## SalePrice ~ GrLivArea + LotArea + HalfBath + LotFrontage + LandSlope +
##     MiscVal + YrSold + X3SsnPorch + MoSold + Utilities
```

```
rmsepara
```

```
## [1] 43321.43
```

With the multiple linear regression, the best model is: SalePrice ~ GrLivArea + LotArea + HalfBath + LotFrontage + LandSlope + X3SsnPorch + Utilities + YrSold + MoSold + MiscVal Test RMSE is : 43321.43

This test RMSE is still quite high for predicting house price.

Conclusion

In this analysis, multiple linear regression, lasso regression, random forest regression, nonlinearity evaluation, and interaction between quantitative variables were explored to enhance test RMSE. Despite using complex models and regression methods, the best performing was a simple regression model with SalePrice ~ GrLivArea. Once the discovery of this, the model was improved using multiple linear regression by implementing automation of adding variables in an order of importance from random forest when the added model resulted better test RMSE. Such method improved test RMSE, but the test RMSE is still high as a prediction model.

Area of Improvement

1. Condensing data to reduce the dimension through autoencoder and PCA then regressing with them could be test.
2. Eliminating many qualitative variables. There are multiple qualitative predictors that have high levels which will conly worsen the model by introducing too many dummy variables. Having many dummy variables exacerbate interpretability even though the model meets regression assumptions.

3. Interaction between quantitative and qualitative varaibles. Since quantitative variable interactions didn't offer much, I suspect this could enhance the test RMSE.
4. Finding more reliable predictors. If bias is the problem, which I suspect since lowering variance methods like lasso and randomforest didn't offer much, the only way to solve this issue is discovering more reliable predictors.