Real Estate Sales Price Prediction

Eric Peterson

July 18, 2017

Overview

Real estate sales price prediction is a hot topic in the field. Driven by the desire for an automated method of estimating home prices, companies such as Zillow, Trulia (now a subsidiary of Zillow), and the National Association of Realtors (NAR), have pursued models that tackle this problem. AVM (automated valuation models) are also prevalent in large scale real estate operations, particularly when dealing with bank owned foreclosures.

This project is an investigation into utilizing machine learning techniques to better predict home prices using the Ames Housing dataset (provided by Kaggle.com).

Training and Test Data

Assuming the proper data files are in the working directory, we can load the already fairly clean data sets into R as data frames easily.

```
library(dplyr)
library(ggplot2)
library(Hmisc)
library(caret)
training <- read.csv("train.csv")
```

Cleaning the Data Set

Now, we need to dig into the features a little and see if we can't determine what we need to adjust and how we are going to deal with missing values.

```
names(training)
```

```
##  [1] "Id"            "MSSubClass"    "MSZoning"      "LotFrontage"
##  [5] "LotArea"       "Street"        "Alley"         "LotShape"
##  [9] "LandContour"   "Utilities"     "LotConfig"     "LandSlope"
## [13] "Neighborhood"  "Condition1"    "Condition2"    "BldgType"
## [17] "HouseStyle"    "OverallQual"   "OverallCond"   "YearBuilt"
## [21] "YearRemodAdd"  "RoofStyle"     "RoofMatl"      "Exterior1st"
## [25] "Exterior2nd"   "MasVnrType"    "MasVnrArea"    "ExterQual"
## [29] "ExterCond"     "Foundation"    "BsmtQual"      "BsmtCond"
## [33] "BsmtExposure"  "BsmtFinType1"  "BsmtFinSF1"    "BsmtFinType2"
## [37] "BsmtFinSF2"    "BsmtUnfSF"     "TotalBsmtSF"   "Heating"
## [41] "HeatingQC"     "CentralAir"    "Electrical"    "X1stFlrSF"
## [45] "X2ndFlrSF"     "LowQualFinSF"  "GrLivArea"     "BsmtFullBath"
## [49] "BsmtHalfBath"  "FullBath"      "HalfBath"      "BedroomAbvGr"
## [53] "KitchenAbvGr"  "KitchenQual"   "TotRmsAbvGrd"  "Functional"
## [57] "Fireplaces"    "FireplaceQu"   "GarageType"    "GarageYrBlt"
## [61] "GarageFinish"  "GarageCars"    "GarageArea"    "GarageQual"
## [65] "GarageCond"    "PavedDrive"    "WoodDeckSF"    "OpenPorchSF"
```

```
## [69] "EnclosedPorch" "X3SsnPorch"    "ScreenPorch"   "PoolArea"
## [73] "PoolQC"        "Fence"         "MiscFeature"   "MiscVal"
## [77] "MoSold"        "YrSold"        "SaleType"      "SaleCondition"
## [81] "SalePrice"
```

First, we take a look at the features themselves (using the training set). There are 81 columns, including the Sales Price, which is what we're building a model to predict. That means we have 80 features to build our model from.

```r
str(training)
```

```
## '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     : 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 NA 51 50 ...
##  $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
##  $ Street       : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Alley        : Factor w/ 2 levels "Grvl","Pave": NA NA NA NA NA NA NA NA NA NA ...
##  $ LotShape     : Factor w/ 4 levels "IR1","IR2","IR3",..: 4 4 1 1 1 1 4 1 4 4 ...
##  $ LandContour  : Factor w/ 4 levels "Bnk","HLS","Low",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Utilities    : Factor w/ 2 levels "AllPub","NoSeWa": 1 1 1 1 1 1 1 1 1 1 ...
##  $ LotConfig    : Factor w/ 5 levels "Corner","CulDSac",..: 5 3 5 1 3 5 5 1 5 1 ...
##  $ LandSlope    : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
##  $ 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   : Factor w/ 4 levels "BrkCmn","BrkFace",..: 2 3 2 3 2 3 4 4 3 3 ...
##  $ MasVnrArea   : int  196 0 162 0 350 0 186 240 0 0 ...
##  $ ExterQual    : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 4 3 4 3 4 4 4 ...
##  $ ExterCond    : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ Foundation   : Factor w/ 6 levels "BrkTil","CBlock",..: 3 2 3 1 3 6 3 2 1 1 ...
##  $ BsmtQual     : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 3 3 4 3 3 1 3 4 4 ...
##  $ BsmtCond     : Factor w/ 4 levels "Fa","Gd","Po",..: 4 4 4 2 4 4 4 4 4 4 ...
##  $ BsmtExposure : Factor w/ 4 levels "Av","Gd","Mn",..: 4 2 3 4 1 4 1 3 4 4 ...
##  $ BsmtFinType1 : Factor w/ 6 levels "ALQ","BLQ","GLQ",..: 3 1 3 1 3 3 3 1 6 3 ...
##  $ BsmtFinSF1   : int  706 978 486 216 655 732 1369 859 0 851 ...
##  $ BsmtFinType2 : Factor w/ 6 levels "ALQ","BLQ","GLQ",..: 6 6 6 6 6 6 6 2 6 6 ...
##  $ 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      : Factor w/ 6 levels "Floor","GasA",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ HeatingQC    : Factor w/ 5 levels "Ex","Fa","Gd",..: 1 1 1 3 1 1 1 1 3 1 ...
##  $ CentralAir   : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ 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 : 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  : Factor w/ 4 levels "Ex","Fa","Gd",..: 3 4 3 3 3 4 3 4 4 4 ...
## $ TotRmsAbvGrd : int  8 6 6 7 9 5 7 7 8 5 ...
## $ Functional   : Factor w/ 7 levels "Maj1","Maj2",..: 7 7 7 7 7 7 7 7 3 7 ...
## $ Fireplaces   : int  0 1 1 1 1 0 1 2 2 2 ...
## $ FireplaceQu  : Factor w/ 5 levels "Ex","Fa","Gd",..: NA 5 5 3 5 NA 3 5 5 5 ...
## $ GarageType   : Factor w/ 6 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 : Factor w/ 3 levels "Fin","RFn","Unf": 2 2 2 3 2 3 2 2 3 2 ...
## $ 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   : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 2 3 ...
## $ GarageCond   : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
## $ PavedDrive   : Factor w/ 3 levels "N","P","Y": 3 3 3 3 3 3 3 3 3 3 ...
## $ 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       : Factor w/ 3 levels "Ex","Fa","Gd": NA NA NA NA NA NA NA NA NA NA ...
## $ Fence        : Factor w/ 4 levels "GdPrv","GdWo",..: NA NA NA NA NA 3 NA NA NA NA ...
## $ MiscFeature  : Factor w/ 4 levels "Gar2","Othr",..: NA NA NA NA NA 3 NA 3 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     : 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 ...
```

That's a little long, but we can already get a feel for the data we're dealing with. One good thing is that some of the data is already set up as factors, which makes life a bit easier. We can also already see that we've got some missing values. We'll need to address those first, before we can build our model. Let's look at one more thing.

```
summary(training)
```

```
##        Id           MSSubClass       MSZoning     LotFrontage    
##  Min.   :   1.0   Min.   : 20.0   C (all):  10   Min.   : 21.00  
##  1st Qu.: 365.8   1st Qu.: 20.0   FV     :  65   1st Qu.: 59.00  
##  Median : 730.5   Median : 50.0   RH     :  16   Median : 69.00  
##  Mean   : 730.5   Mean   : 56.9   RL     :1151   Mean   : 70.05  
##  3rd Qu.:1095.2   3rd Qu.: 70.0   RM     : 218   3rd Qu.: 80.00  
##  Max.   :1460.0   Max.   :190.0                  Max.   :313.00  
##                                                  NA's   :259     
##     LotArea        Street      Alley      LotShape  LandContour
```

```
##  Min.   :  1300   Grvl:   6   Grvl:  50   IR1:484   Bnk:  63
##  1st Qu.:  7554   Pave:1454   Pave:  41   IR2: 41   HLS:  50
##  Median :  9478               NA's:1369   IR3: 10   Low:  36
##  Mean   : 10517                           Reg:925   Lvl:1311
##  3rd Qu.: 11602
##  Max.   :215245
##
##   Utilities      LotConfig    LandSlope   Neighborhood   Condition1
##  AllPub:1459   Corner : 263   Gtl:1382   NAmes  :225   Norm   :1260
##  NoSeWa:   1   CulDSac:  94   Mod:  65   CollgCr:150   Feedr  :  81
##                FR2    :  47   Sev:  13   OldTown:113   Artery :  48
##                FR3    :   4              Edwards:100   RRAn   :  26
##                Inside :1052              Somerst: 86   PosN   :  19
##                                          Gilbert: 79   RRAe   :  11
##                                          (Other):707   (Other):  15
##    Condition2     BldgType      HouseStyle   OverallQual
##  Norm   :1445   1Fam  :1220   1Story :726   Min.   : 1.000
##  Feedr  :   6   2fmCon:  31   2Story :445   1st Qu.: 5.000
##  Artery :   2   Duplex:  52   1.5Fin :154   Median : 6.000
##  PosN   :   2   Twnhs :  43   SLvl   : 65   Mean   : 6.099
##  RRNn   :   2   TwnhsE: 114   SFoyer : 37   3rd Qu.: 7.000
##  PosA   :   1                 1.5Unf : 14   Max.   :10.000
##  (Other):   2                 (Other): 19
##   OverallCond      YearBuilt     YearRemodAdd    RoofStyle
##  Min.   :1.000   Min.   :1872   Min.   :1950   Flat   :  13
##  1st Qu.:5.000   1st Qu.:1954   1st Qu.:1967   Gable  :1141
##  Median :5.000   Median :1973   Median :1994   Gambrel:  11
##  Mean   :5.575   Mean   :1971   Mean   :1985   Hip    : 286
##  3rd Qu.:6.000   3rd Qu.:2000   3rd Qu.:2004   Mansard:   7
##  Max.   :9.000   Max.   :2010   Max.   :2010   Shed   :   2
##
##     RoofMatl     Exterior1st   Exterior2nd    MasVnrType    MasVnrArea
##  CompShg:1434   VinylSd:515   VinylSd:504   BrkCmn : 15   Min.   :   0.0
##  Tar&Grv:  11   HdBoard:222   MetalSd:214   BrkFace:445   1st Qu.:   0.0
##  WdShngl:   6   MetalSd:220   HdBoard:207   None   :864   Median :   0.0
##  WdShake:   5   Wd Sdng:206   Wd Sdng:197   Stone  :128   Mean   : 103.7
##  ClyTile:   1   Plywood:108   Plywood:142   NA's   :  8   3rd Qu.: 166.0
##  Membran:   1   CemntBd: 61   CmentBd: 60                 Max.   :1600.0
##  (Other):   2   (Other):128   (Other):136                 NA's   :8
##  ExterQual ExterCond  Foundation  BsmtQual   BsmtCond    BsmtExposure
##  Ex: 52    Ex:   3   BrkTil:146   Ex  :121   Fa  :  45   Av  :221
##  Fa: 14    Fa:  28   CBlock:634   Fa  : 35   Gd  :  65   Gd  :134
##  Gd:488    Gd: 146   PConc :647   Gd  :618   Po  :   2   Mn  :114
##  TA:906    Po:   1   Slab  : 24   TA  :649   TA  :1311   No  :953
##            TA:1282   Stone :  6   NA's: 37   NA's:  37   NA's: 38
##                      Wood  :  3
##
##  BsmtFinType1   BsmtFinSF1     BsmtFinType2   BsmtFinSF2
##  ALQ :220     Min.   :   0.0   ALQ :  19    Min.   :   0.00
##  BLQ :148     1st Qu.:   0.0   BLQ :  33    1st Qu.:   0.00
##  GLQ :418     Median : 383.5   GLQ :  14    Median :   0.00
##  LwQ : 74     Mean   : 443.6   LwQ :  46    Mean   :  46.55
##  Rec :133     3rd Qu.: 712.2   Rec :  54    3rd Qu.:   0.00
##  Unf :430     Max.   :5644.0   Unf :1256    Max.   :1474.00
```

```
##  NA's: 37                     NA's:  38
##    BsmtUnfSF       TotalBsmtSF      Heating     HeatingQC CentralAir
##  Min.   :   0.0   Min.   :   0.0   Floor:   1   Ex:741    N:  95
##  1st Qu.: 223.0   1st Qu.: 795.8   GasA :1428   Fa: 49    Y:1365
##  Median : 477.5   Median : 991.5   GasW :  18   Gd:241
##  Mean   : 567.2   Mean   :1057.4   Grav :   7   Po:  1
##  3rd Qu.: 808.0   3rd Qu.:1298.2   OthW :   2   TA:428
##  Max.   :2336.0   Max.   :6110.0   Wall :   4
##
##  Electrical     X1stFlrSF      X2ndFlrSF     LowQualFinSF
##  FuseA:  94   Min.   : 334   Min.   :   0   Min.   :  0.000
##  FuseF:  27   1st Qu.: 882   1st Qu.:   0   1st Qu.:  0.000
##  FuseP:   3   Median :1087   Median :   0   Median :  0.000
##  Mix  :   1   Mean   :1163   Mean   : 347   Mean   :  5.845
##  SBrkr:1334   3rd Qu.:1391   3rd Qu.: 728   3rd Qu.:  0.000
##  NA's :   1   Max.   :4692   Max.   :2065   Max.   :572.000
##
##    GrLivArea     BsmtFullBath     BsmtHalfBath       FullBath
##  Min.   : 334   Min.   :0.0000   Min.   :0.00000   Min.   :0.000
##  1st Qu.:1130   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:1.000
##  Median :1464   Median :0.0000   Median :0.00000   Median :2.000
##  Mean   :1515   Mean   :0.4253   Mean   :0.05753   Mean   :1.565
##  3rd Qu.:1777   3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:2.000
##  Max.   :5642   Max.   :3.0000   Max.   :2.00000   Max.   :3.000
##
##     HalfBath       BedroomAbvGr    KitchenAbvGr   KitchenQual
##  Min.   :0.0000   Min.   :0.000   Min.   :0.000   Ex:100
##  1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:1.000   Fa: 39
##  Median :0.0000   Median :3.000   Median :1.000   Gd:586
##  Mean   :0.3829   Mean   :2.866   Mean   :1.047   TA:735
##  3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:1.000
##  Max.   :2.0000   Max.   :8.000   Max.   :3.000
##
##   TotRmsAbvGrd    Functional    Fireplaces    FireplaceQu   GarageType
##  Min.   : 2.000   Maj1:  14   Min.   :0.000   Ex  : 24    2Types :  6
##  1st Qu.: 5.000   Maj2:   5   1st Qu.:0.000   Fa  : 33    Attchd :870
##  Median : 6.000   Min1:  31   Median :1.000   Gd  :380    Basment: 19
##  Mean   : 6.518   Min2:  34   Mean   :0.613   Po  : 20    BuiltIn: 88
##  3rd Qu.: 7.000   Mod :  15   3rd Qu.:1.000   TA  :313    CarPort:  9
##  Max.   :14.000   Sev :   1   Max.   :3.000   NA's:690    Detchd :387
##                   Typ :1360                               NA's   : 81
##   GarageYrBlt   GarageFinish   GarageCars      GarageArea     GarageQual
##  Min.   :1900   Fin :352     Min.   :0.000   Min.   :   0.0   Ex  :   3
##  1st Qu.:1961   RFn :422     1st Qu.:1.000   1st Qu.: 334.5   Fa  :  48
##  Median :1980   Unf :605     Median :2.000   Median : 480.0   Gd  :  14
##  Mean   :1979   NA's: 81     Mean   :1.767   Mean   : 473.0   Po  :   3
##  3rd Qu.:2002                3rd Qu.:2.000   3rd Qu.: 576.0   TA  :1311
##  Max.   :2010                Max.   :4.000   Max.   :1418.0   NA's:  81
##  NA's   :81
##  GarageCond  PavedDrive   WoodDeckSF      OpenPorchSF     EnclosedPorch
##  Ex  :   2   N:  90     Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
##  Fa  :  35   P:  30     1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.00
##  Gd  :   9   Y:1340     Median :  0.00   Median : 25.00   Median :  0.00
##  Po  :   7              Mean   : 94.24   Mean   : 46.66   Mean   : 21.95
```

```
##  TA  :1326              3rd Qu.:168.00   3rd Qu.: 68.00   3rd Qu.:  0.00
##  NA's:  81              Max.   :857.00   Max.   :547.00   Max.   :552.00
##
##   X3SsnPorch      ScreenPorch        PoolArea        PoolQC
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.000   Ex  :   2
##  1st Qu.:  0.00   1st Qu.:  0.00   1st Qu.:  0.000   Fa  :   2
##  Median :  0.00   Median :  0.00   Median :  0.000   Gd  :   3
##  Mean   :  3.41   Mean   : 15.06   Mean   :  2.759   NA's:1453
##  3rd Qu.:  0.00   3rd Qu.:  0.00   3rd Qu.:  0.000
##  Max.   :508.00   Max.   :480.00   Max.   :738.000
##
##   Fence      MiscFeature    MiscVal             MoSold
##  GdPrv:  59   Gar2:   2   Min.   :    0.00   Min.   : 1.000
##  GdWo :  54   Othr:   2   1st Qu.:    0.00   1st Qu.: 5.000
##  MnPrv: 157   Shed:  49   Median :    0.00   Median : 6.000
##  MnWw :  11   TenC:   1   Mean   :   43.49   Mean   : 6.322
##  NA's :1179   NA's:1406   3rd Qu.:    0.00   3rd Qu.: 8.000
##                            Max.   :15500.00   Max.   :12.000
##
##      YrSold        SaleType    SaleCondition    SalePrice
##  Min.   :2006   WD     :1267   Abnorml: 101   Min.   : 34900
##  1st Qu.:2007   New    : 122   AdjLand:   4   1st Qu.:129975
##  Median :2008   COD    :  43   Alloca :  12   Median :163000
##  Mean   :2008   ConLD  :   9   Family :  20   Mean   :180921
##  3rd Qu.:2009   ConLI  :   5   Normal :1198   3rd Qu.:214000
##  Max.   :2010   ConLw  :   5   Partial: 125   Max.   :755000
##                 (Other):   9
```

Immediately, we can see that a lot of the missing values are probably linked to the absence of another feature, such as alley access, a garage, or a basement (this is also reflected in the data code book). We'll want to create a factor level that corresponds to None (or not applicable) for those cases. In this case, I'll show one and do the rest with echo off.

```r
new_levels <- levels(training$Alley)
new_levels[length(new_levels)+1] <- "NA"
training$Alley <- factor(training$Alley, levels = new_levels)
training$Alley[is.na(training$Alley)] <- "NA"
```

With that taken care of, we can look at the other features that have missing data.

```r
colSums(sapply(training, is.na))
```

```
##            Id    MSSubClass      MSZoning   LotFrontage       LotArea
##             0             0             0           259             0
##        Street         Alley      LotShape   LandContour     Utilities
##             0             0             0             0             0
##     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             8             0             0             0
##      BsmtQual      BsmtCond  BsmtExposure  BsmtFinType1    BsmtFinSF1
##             0             0             0             0             0
```

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

So, we have 3 features with missing values left, LotFrontage, MasVnrArea (the area of a masonry veneer), and GarageYrBlt. Two of these are easy to explain, 8 homes probably have no veneer and 81 have no garage, so we can justify setting these to 0. Lot Frontage is interesting, but, even here, we're probably dealing with condominiums and/or town houses that have minimal to no frontage, so we're going to set those to 0 as well (alternately, we could impute the data in some way).

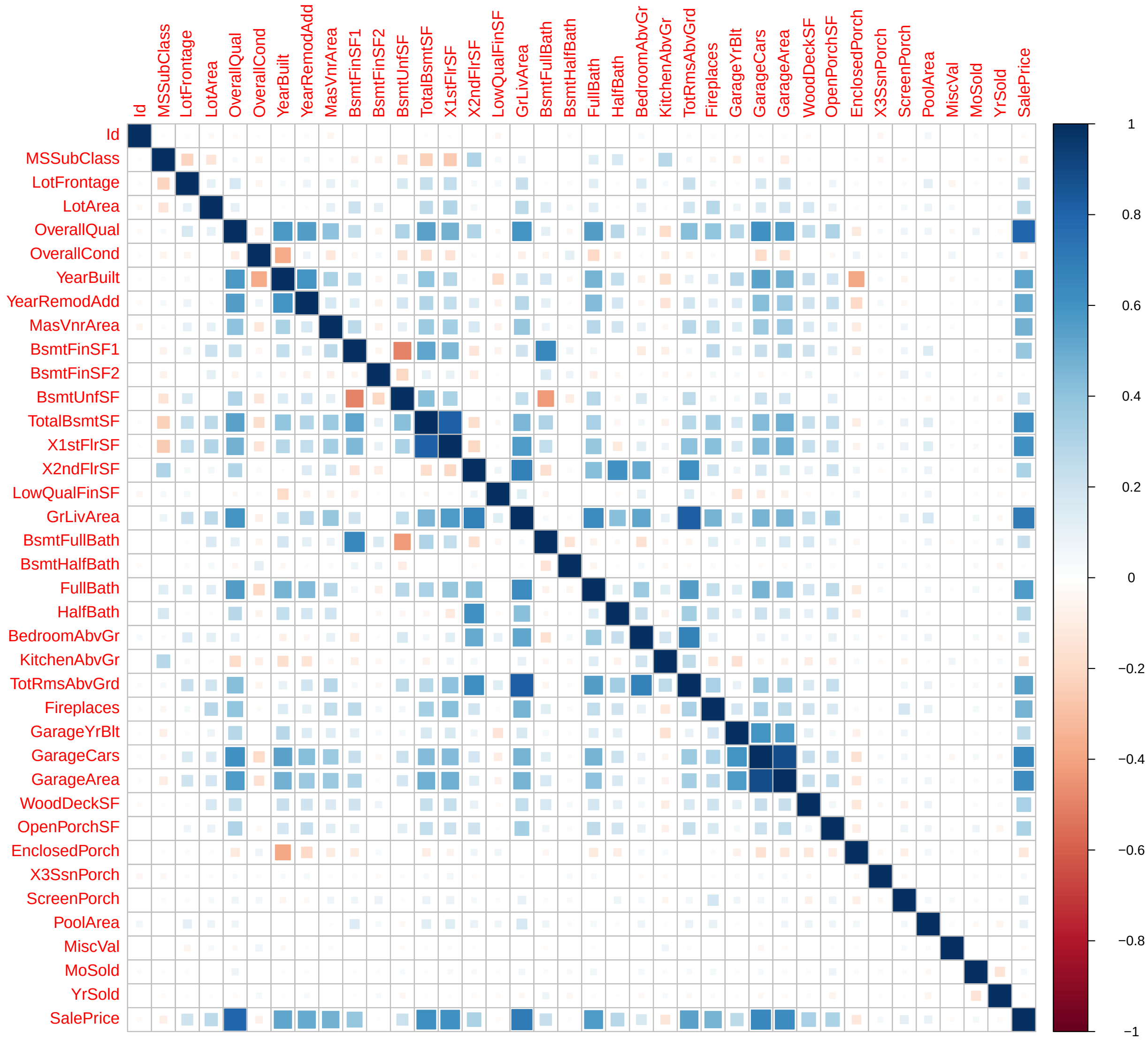
```
training$GarageYrBlt[is.na(training$GarageYrBlt)] <- 0
training$MasVnrArea[is.na(training$MasVnrArea)] <- 0
training$LotFrontage[is.na(training$LotFrontage)] <- 0
```

Our data is now free of missing values. We may decide to do more to it, but that will be after some analysis.

Exploritory Analysis of the Data Set

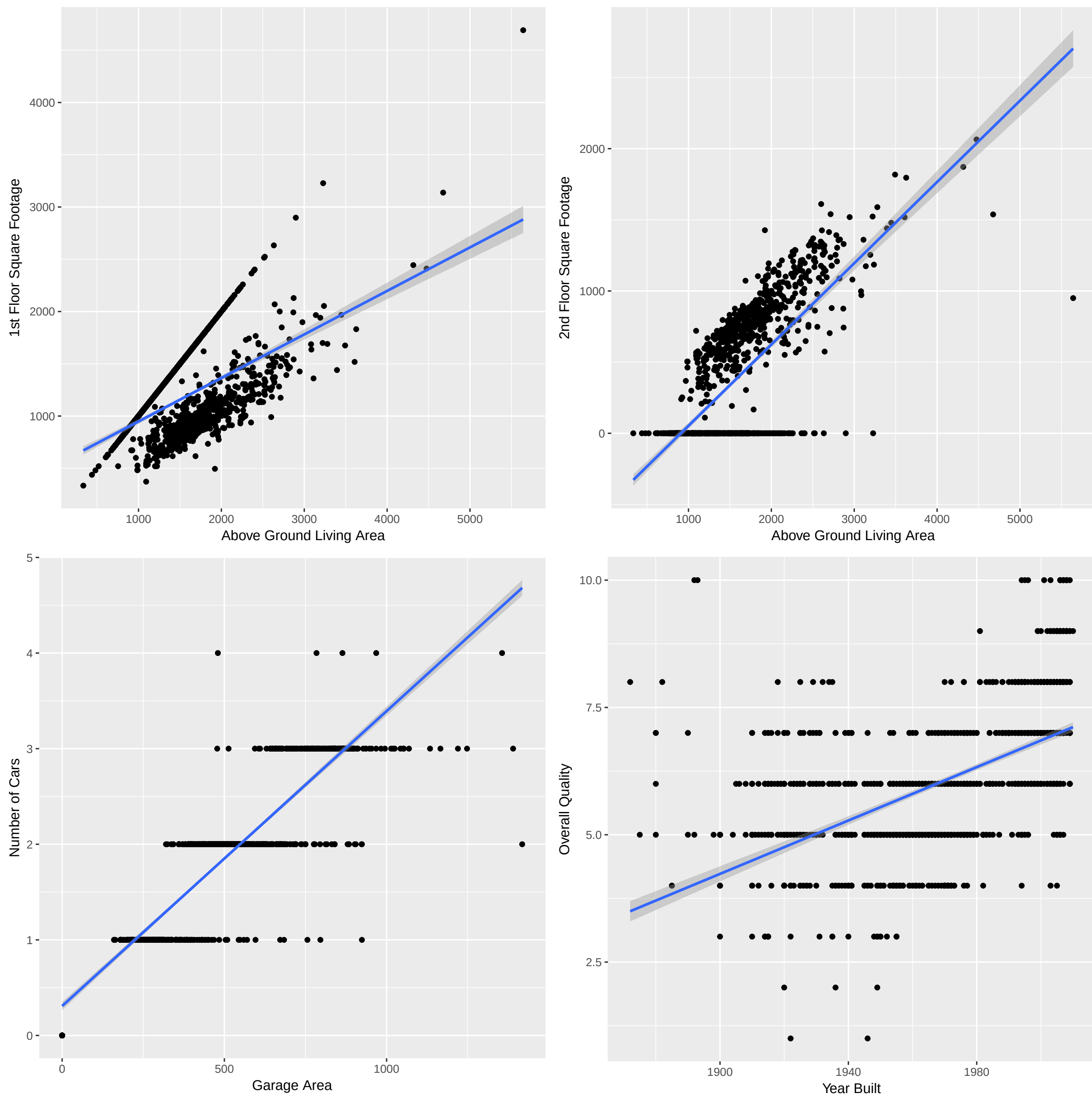
Let's look at some correlations. This will give us some idea of where we have features that are linked in some way and also give us some insight into what features have the greatest effect on the sales price.

```
library(corrplot)
# find numeric columns in the data frame and store in logical
num_cols <- sapply(training, is.numeric)

#select out the numeric columns
num_train <- training[ , num_cols]

#find and plot the correlation matrix
correlations <- cor(num_train, use = "everything")
corrplot(correlations, method = "square")
```



We immediately see some interesting things. First off, we see some features that correlate highly. Most are intuitive, as we see with Garage Area and Number of Cars and living area.

```
library(gridExtra)

g1 <- qplot(GrLivArea, X1stFlrSF, data = training, geom = c("point", "smooth"), method = "lm", xlab = "

g2 <- qplot(GrLivArea, X2ndFlrSF, data = training, geom = c("point", "smooth"), method = "lm", xlab = "

g3 <- qplot(GarageArea, GarageCars, data = training, geom = c("point", "smooth"), method = "lm", xlab =

g4 <- qplot(YearBuilt, OverallQual, data = training, geom = c("point", "smooth"), method = "lm", xlab =

grid.arrange(g1, g2, g3, g4, ncol = 2)
```



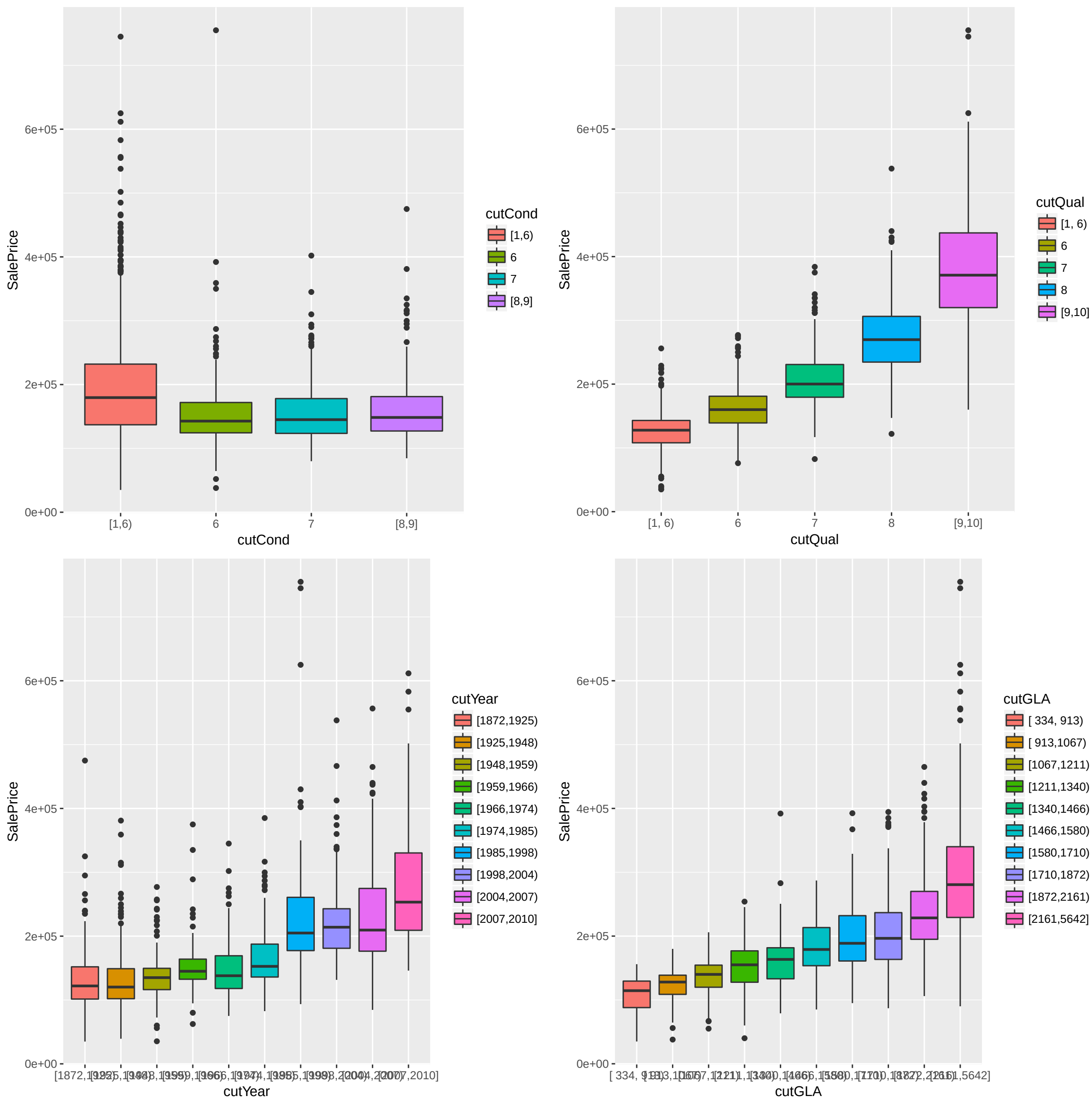
It's clear that as the garage area increases, the number of cars tends to as well. We illustrate it with a simple linear fit that only starts to break down at the extremes when there are few samples.

Living area is a little more interesting. There's clearly a split in the data, although it's fairly easy to explain. Some houses are only one floor, so the 1st Floor Area will be the same as the Above Ground Living Area. Similarly, if the house only has one floor, 2nd Floor Area will be 0. Since both of these values track very closely with total above ground living area, we can make an argument to condense or eliminate them when we build our model.

Year and overall quality are also correlated, although a bit weaker than the others. Let's look at some of our features that correlate highly with the sales price.

```
#use Cut2() from Hmisc to cut numerical features into bins
cutQual <- cut2(training$OverallQual, g = 10)
cutCond <- cut2(training$OverallCond, g = 10)
cutYear <- cut2(training$YearBuilt, g = 10)
```

```r
cutGLA  <- cut2(training$GrLivArea, g = 10)

bp1 <- qplot(cutCond, SalePrice, data = training, fill = cutCond, geom = c("boxplot"))

bp2 <- qplot(cutQual, SalePrice, data = training, fill = cutQual, geom = c("boxplot"))

bp3 <- qplot(cutYear, SalePrice, data = training, fill = cutYear, geom = c("boxplot"))

bp4 <- qplot(cutGLA, SalePrice, data = training, fill = cutGLA, geom = c("boxplot"))

grid.arrange(bp1, bp2, bp3, bp4, ncol = 2)
```

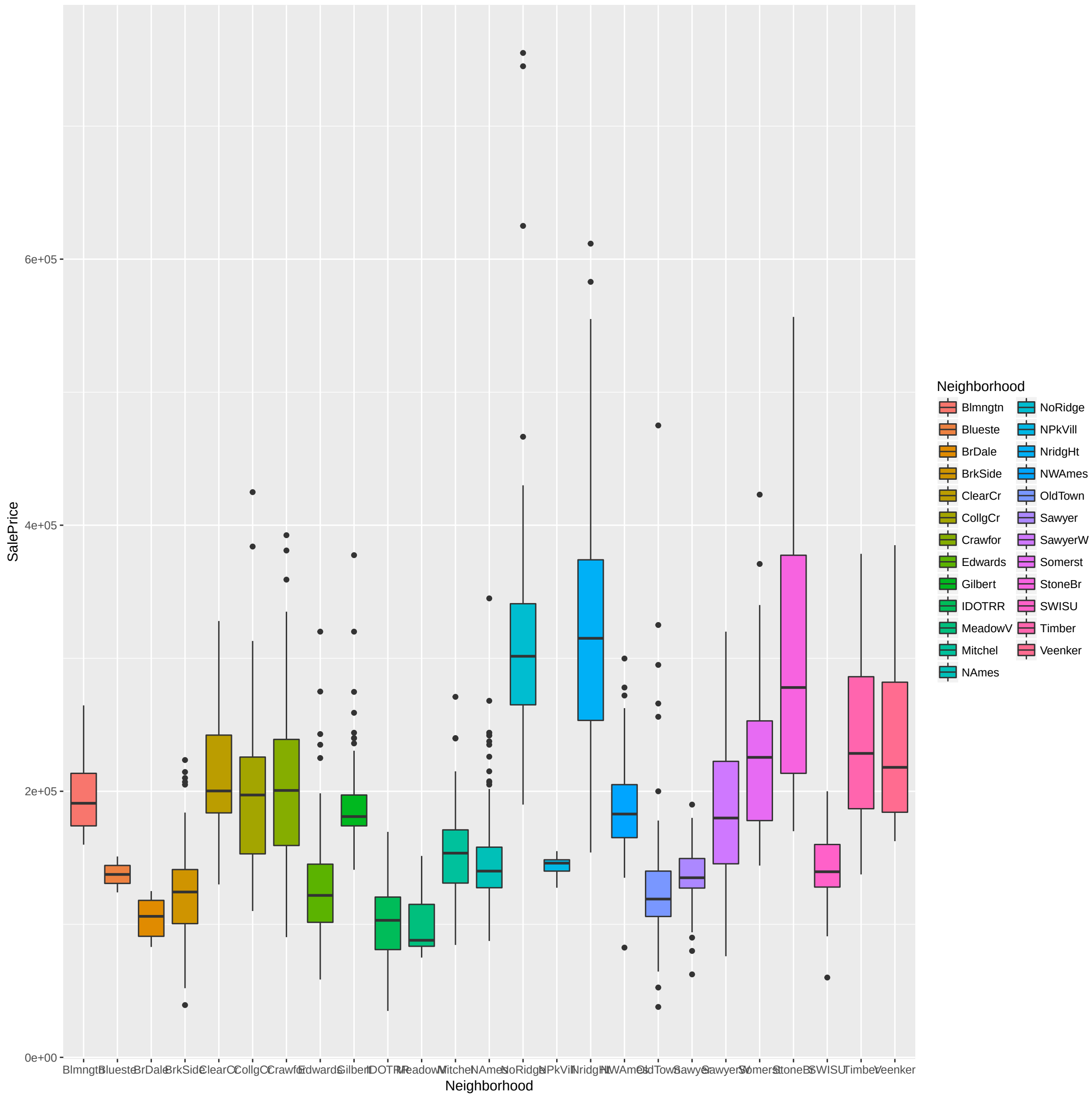


Surprisingly, overall condition is very poorly correlated to sales price. Looking at the data, most values fall in the 1-6 range, but even accounting for that, there's not a huge effect. This could also be a flaw in the rating system used to determine home condition.

Quality and above ground living area show strong effects while the year built seems to break down in 30 year increments.

One factor variable should also have a fairly large effect on sales price and that's neighborhood. Location is one of the biggest predictors of house pricing.

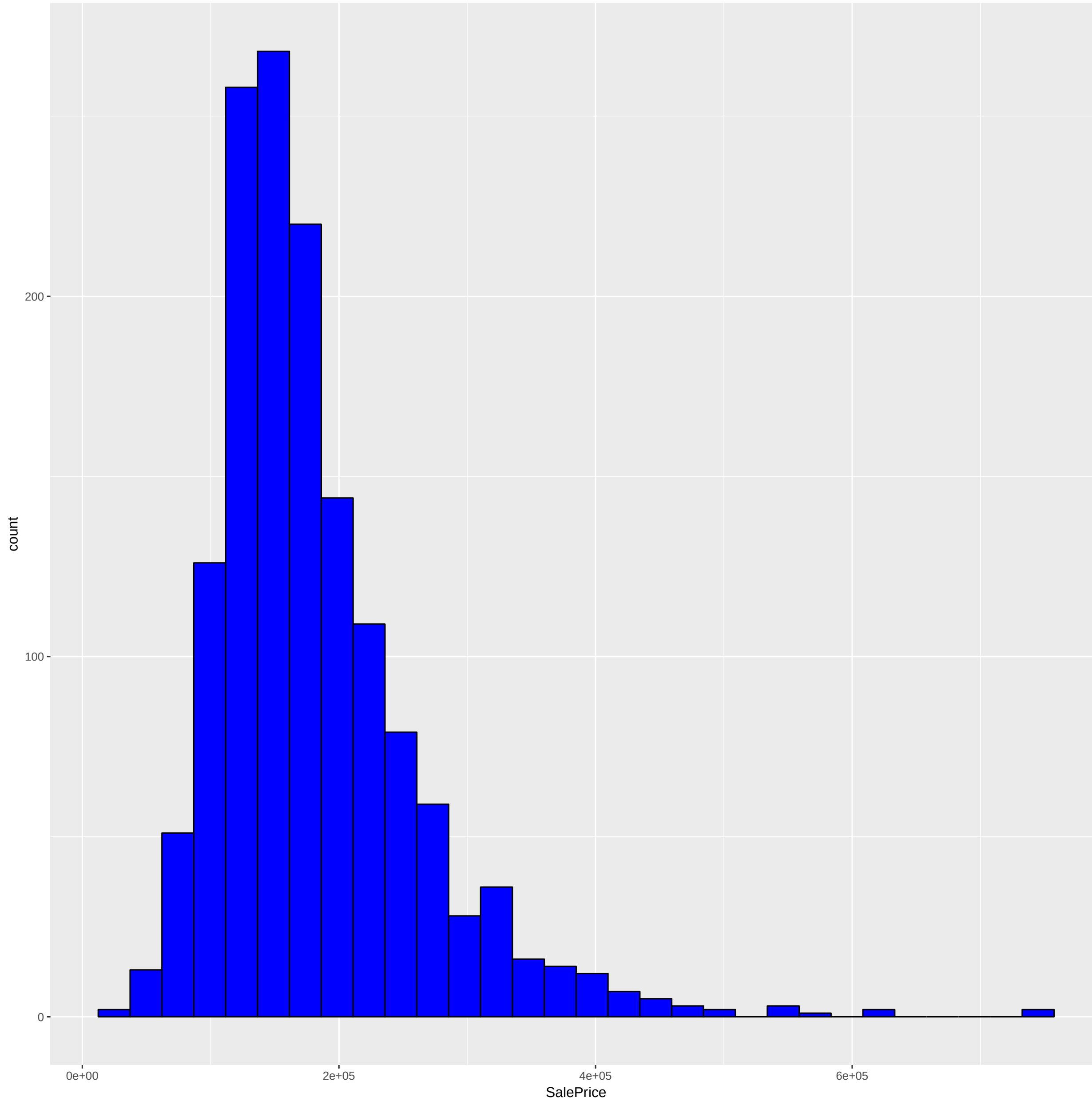
```r
qplot(Neighborhood, SalePrice, data = training, fill = Neighborhood, geom = c("boxplot"))
```



Sure enough, we can see that certain neighborhoods are more expensive than others. Before we put together a model, we want to look at one more thing; the distribution of sales prices.

```r
qplot(SalePrice, data = training, fill = I("blue"), col = I("black"))
```

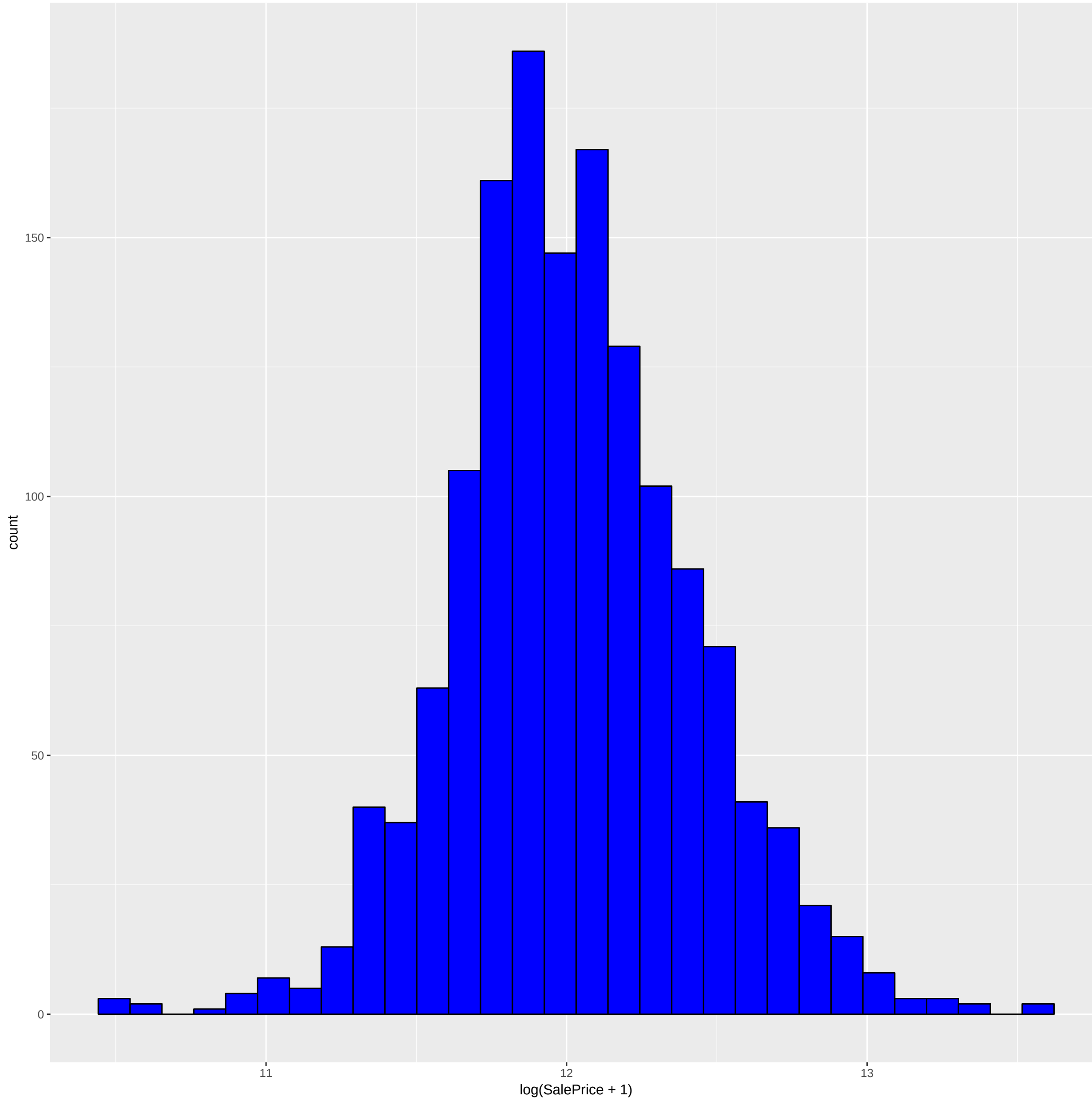
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Looks like the sale prices are skewed. We can address this by taking the log of the price.

```
qplot(log(SalePrice + 1), data = training, fill = I("blue"), col = I("black"))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Now we can start looking at models.

```
control <- trainControl(method = "cv", number = 10)

lm_model <- train(SalePrice ~ ., method = "lm", data = training, trControl = control)

lm_model$results
```

```
##   intercept     RMSE  Rsquared   RMSESD RsquaredSD
## 1      TRUE 46489.57 0.7195149 20604.24  0.1687289
```

```
print(lm_model)
```

```
## Linear Regression
##
## 1460 samples
```

```
##   80 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1313, 1314, 1314, 1313, 1315, 1313, ...
## Resampling results:
##
##   RMSE      Rsquared
##   46489.57  0.7195149
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

We now have a simple linear model of the data. The summary is only showing us one model run, so we can't go off of those values. We also used caret to perform a k-fold cross validation with 10 subsets. We could manually separate out a training and test set, but k-fold should give us better results. As you can see from the standard deviations of the RMSE and RSquared, the trade off for k-fold is high variance.

Let's try with log(SalePrice + 1).

```
control <- trainControl(method = "cv", number = 10)

lm_model_log <- train(log(SalePrice + 1) ~ ., method = "lm", data = training, trControl = control)

print(lm_model_log)
```

```
## Linear Regression
##
## 1460 samples
##   80 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1314, 1314, 1315, 1315, 1315, 1312, ...
## Resampling results:
##
##   RMSE       Rsquared
##   0.1739888  0.8092079
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
lm_model_log$results
```

```
##   intercept      RMSE  Rsquared    RMSESD RsquaredSD
## 1      TRUE 0.1739888 0.8092079 0.0816394  0.1619918
```

Here, model accuracy has improved and variance, while still high, is a little better. This is a little tricky, as we did do a data transformation, so comparing values is of limited utility. Let's see what happens if we only use a selection of the data. As we discussed above, we'll only use the data that shows strong correlation to the sales price and throw out redundant data.

```
control <- trainControl(method = "cv", number = 10)

lm_model_log_s <- train(log(SalePrice + 1) ~ OverallQual + YearBuilt + YearRemodAdd + MasVnrArea + Total

lm_model_log_s$results
```

```
##   intercept      RMSE  Rsquared    RMSESD RsquaredSD
## 1      TRUE 0.1583382 0.8458342 0.0481854 0.08527063
```

```r
print(lm_model_log_s)
```

```
## Linear Regression
##
## 1460 samples
##   17 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1316, 1312, 1314, 1314, 1314, 1315, ...
## Resampling results:
##
##   RMSE       Rsquared
##   0.1583382  0.8458342
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Our accuracy gets better again while variance drops from the last model (since we used the same transformation, this comparison is much more applicable). That's not a bad trade off and it took less variables to do it. Of course, this is assuming a linear relationship between the features and the Sales Price, which may very well not be the case. With that in mind, let's try something completely different, a random forest model.

```r
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.4.1
```

```r
set.seed(3218)
control <- trainControl(method = "cv", number = 5)


mtry <- sqrt(ncol(training))
tunegrid <- expand.grid(.mtry = mtry)

rf_model <- train(SalePrice ~., data = training, method = "rf", trControl = control, tuneGrid = tunegri

print(rf_model)
```

```
## Random Forest
##
## 1460 samples
##   80 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 1167, 1169, 1167, 1168, 1169
## Resampling results:
##
##   RMSE      Rsquared
##   31941.45  0.8610768
##
## Tuning parameter 'mtry' was held constant at a value of 9
```

```r
print(rf_model$finalModel)
```

```
##
## Call:
```

```
##  randomForest(x = x, y = y, mtry = param$mtry, proximity = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 9
##
##           Mean of squared residuals: 1032843795
##                     % Var explained: 83.62
```

We're using 5 folds in order to speed the calculations up, as random forest takes some processor time. We're restricting it to the square root of the number of features for variables at each split. In general, this gives us a more accurate model, which makes sense since we're using a large number of features to make our predictions. Let's see what we get when we use log(SalePrice + 1) as our target.

```
set.seed(3218)
control <- trainControl(method = "cv", number = 5)


mtry <- sqrt(ncol(training))
tunegrid <- expand.grid(.mtry = mtry)

rf_model <- train(log(SalePrice + 1) ~., data = training, method = "rf", trControl = control, tuneGrid =

print(rf_model)
```

```
## Random Forest
##
## 1460 samples
##   80 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 1167, 1169, 1167, 1168, 1169
## Resampling results:
##
##   RMSE       Rsquared
##   0.1541093  0.8704845
##
## Tuning parameter 'mtry' was held constant at a value of 9
```

```
print(rf_model$finalModel)
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry, proximity = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 9
##
##           Mean of squared residuals: 0.02332326
##                     % Var explained: 85.37
```