Predicting House Prices From Real-Estate Data

Eric Murphy

5/1/2019

Table of Contents

A. Preface - The 4D Data Science Lifecycle Framework

1. Define - What Problem to Solve : KPIs to Impact
2. Discover
    - Goals for Effective Data Exploration
        - Obtain/Load data
        - Clean data
        - Explore data
        - Establish baseline outcomes
        - Hypothesize solutions
3. Develop
    - Feature Selection
    - Create models
    - Model Testing
    - Model Selection
4. Deploy
    - Solution
    - Measure Model Effectiveness on KPIs

B. Descriptive Statistics, Adequacy Tests, & Feature Selection Methods

1. Introduction - Define
    - Defining our goal - KPIs to Impact
    - Overview of the Dataset
    - Motivations for Analysis
    - Description of Dataset
        - Descriptive Statistics of Missing & Incomplete Data
        - Descriptive Statistics for Discrete & Continuous Variables
        - Identifying Multicollinearity Early (VIF, PCA/PCR)
    - Questions to Consider

2. Exploratory Data Analysis - Discover
    - Visualizing and describing the Response Variable (SalePrice)
        - Measures of Central Tendency (mean, median, mode, outliers)
            - Why using a trimmed mean is ideal
        - Measures of Dispersion (variance, standard deviation, IQR)
        - Measures of Symmetry (skewness, kurtosis)
    - Data Cleaning
        - Checking the data for completeness
        - Imputing missing data
        - Variable Encoding & Factorization

C. Modeling & Methodology - Develop

1. Data Preprocessing Feature Selection Methods
    - Filter Methods

```
      - Pearson Correlation Matrix
   - Embedded Methods
      - LASSO Regression
      - Ridge Regression
      - Random Forest
2. Comparison Tests
      - k-fold Cross Validation
```

D. Final Models - Deploy

```
1. Final Model Summary
   - Tuning Parameters
   - Training Error
   - Prediction Error Estimate
   - Important Features
2. Response Estimate KPIs
   - Parameter Estimations
   - Confidence Intervals
   - Prediction Intervals
   - p-values
```

E. Discussion of Results

```
1. Conclusions
   - Final Accuracy
   - Limitations of analysis
```

##Define - Stating the Problem, Goals, and Overview of our Dataset

###Goal: To define our problem to solve

###Problem: What features best explain the price of a house? Can we then use the best features to predict the sale price of any house in Ames, Iowa

###Description: The Ames Iowa data set describes the sale of individual residential property in Ames, Iowa from 2006 to 2010. The data set contains 2930 observations and a sizeable number of explanatory variables, of which 23 are categorical variables, 23 are ordered variables, 14 are discretized, and 20 are continuous. Upon data cleaning and some imputation of extraneous variables, 80 variables remained that were directly related to the real estate in question that focus on the quality and quantity of many physical attributes of the property.

The 14 discrete variables often quantify the number of items occurring in the house, such as number of kitchens, bedrooms, and full and half-bathrooms, and above ground living areas of the home. The 20 continuous variables pertain to various size dimensions for each observation, usually in square-footage. In addition to the average lot size and total square footage found on most common home listings, other more specific variables are quantified in the data set, such as area measurements on certain rooms such as basements. The primary living area, and even patio space is reduced into individual categories based on quality and type. Remodeling dates are also recorded, which is practical information in house hunting. They range from two to 28 classes with the least described being STREET (gravel or paved) and the often-described being NEIGHBORHOOD. The nominal variables identify various types of dwellings, garages, materials, and environmental conditions while the ordinal variables qualitatively rate various items within the property.

###Motivation for Analysis: Personally, I found interest in this dataset because I am or will soon be actively interested in purchasing a home someday. I found that the analysis would prove to be useful to me as a current student for learning how to think, as well as one day being a data practitioner that navigates the home-buying process. As it would be, most of the variables are the kind of information that the average home buyer would want to know about a potential property, that answer direct and indirect looming questions

such as: 1. When was it built / is it too old to maintain long-term? 2. How big is the lot / is it too small relative to the price? 3. How many square feet of living space is in the dwelling / where's that space used most or least? 4. How many full & half bathrooms are there / will my housemates have to share one, if we have guests will there be a dedicated guest bathroom?

Intuitively, these are useful variables that influence our natural decision-making process. In aggregating information we value highest, we naturally decide if the features justify the price.

###Questions that deserve answers: The genesis for most questions for this project have their roots in human's ability to intuitively derive value from multivariate analysis in day to day life. As such, I primarily explore solutions to four questions: 1. Which variables explain most of the variance and fit of the predictions? Do the most important variables naturally make sense? 2. In doing our analysis, should we exclude highly suspect outliers? If the data is skewed, should we transform it or leave the distribution as-is? Is there regression towards the mean if we leave outliers in the data? 3. Which algorithm predicts best? I will explore Principal Component Regression, and Random Forest methods. Do any methods translate easily to natural intuitive processes? 4. Among discovered correlations, which ones are naturally unintuitive? Does this analysis provide scalable insight?

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```
## Registered S3 method overwritten by 'rvest':
##   method            from
##   read_xml.response xml2
```

```
## -- Attaching packages ---------------------------------------------------------------------- tidyverse 1
```

```
## v ggplot2 3.1.1       v purrr   0.3.2
## v tibble  2.1.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------------------------- tidyverse_conflic
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## Loaded gbm 2.1.5
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
## Registered S3 method overwritten by 'tree':
##   method     from
##   print.tree cli
```

```
##
## Attaching package: 'xgboost'
```

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

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'

## The following object is masked from 'package:tidyr':
##
##     expand

## Loading required package: foreach

##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when

## Loaded glmnet 2.0-16

## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

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

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

## corrplot 0.84 loaded

##
## Attaching package: 'pls'

## The following object is masked from 'package:corrplot':
##
##     corrplot

## The following object is masked from 'package:stats':
##
##     loadings

## Loading required package: carData

##
## Attaching package: 'car'

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

## The following object is masked from 'package:purrr':
##
##     some

## Loading required package: lattice

##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:pls':
##
##     R2

## The following object is masked from 'package:purrr':
##
##     lift

##
## Attaching package: 'gridExtra'

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

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

##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor

## Loading required package: plyr

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##
## Attaching package: 'plyr'

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

## The following object is masked from 'package:purrr':
##
##     compact

##
## Attaching package: 'psych'

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

## The following object is masked from 'package:car':
##
##     logit
```

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

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

After loading in relevant libraries and feeding in training and testing data from the starting 2nd row of the data (to make train and test the same size), we find that the training set has 1,460 observations across 81 features.

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

Our test data has 1,459 observations spread out over the same features minus the response variable SalePrice, which is expected.

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

Discovering Insights

Let's plot a histogram of the SalePrice distribution.

Here are all the features we have to work with in the training set. The test set is the same set of features without the last feature, our response variable SalePrice.

```
##  [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"
```

#Data Cleaning Step

#Since the IDs do not contribute anything to model influence, we will omit these from the training and testing data. If we need them again, we can call them from a test_label variable-object created here.

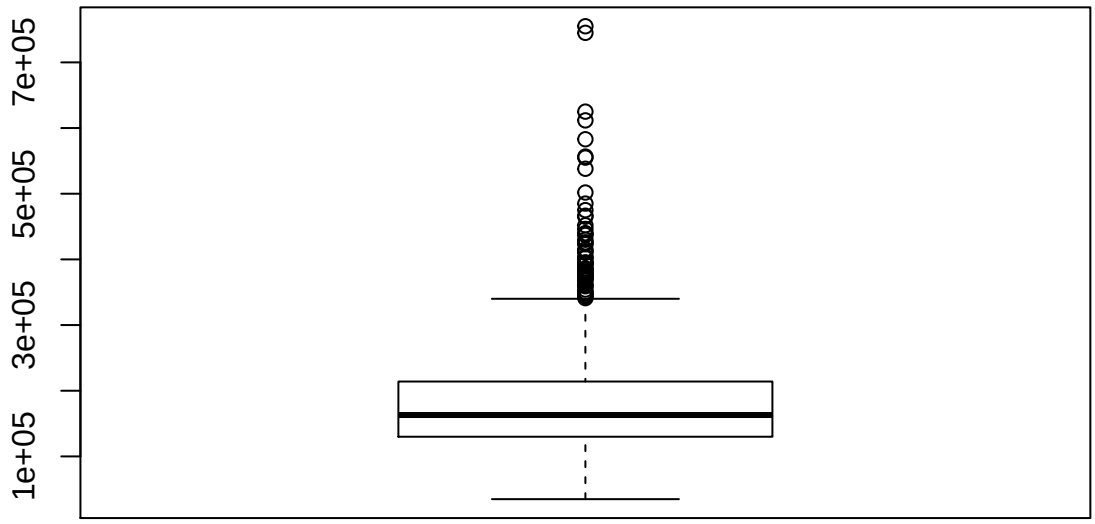
We now revisualize the dimensions of the data, and name listing, just to make sure.

```
## [1] 1460   80
```

```
## [1] 1459   79
```

Here are some descriptive statistics about our response variable SalePrice: Data is skewed highly positively skewed (Skew >1, = 1.88) Population Mean = $189,921 Standard Deviation = $79,442.50 First Quartile/25th %ile = $129,975 Median/50th %ile = $163,000 Third Quartile/75th %ile = $214,000 IQR = Third Quartle - First Quartile = $84,025 Minimum Sale Price= 34,900 and Highest Sale Price = $755,000 Kurtosis = 6.5 (Normal is 3, so this is curve is very steep) which implies that the Mode is very large, which implies that the mean value is observed a lot. Outlier Detection (Traditional): Mean + [3 x Stdev] = SalePrice > $428,249 (19)

```
##    vars    n     mean      sd median  trimmed     mad   min    max  range
## X1    1 1460 180921.2 79442.5 163000 170783.3 56338.8 34900 755000 720100
##    skew kurtosis      se
## X1 1.88      6.5 2079.11
```



I'm curious to see how large the outliers are, so I list the prices of them.

```
##  [1] 345000 385000 438780 383970 372402 412500 501837 475000 386250 403000
## [11] 415298 360000 375000 342643 354000 377426 437154 394432 426000 555000
## [21] 440000 380000 374000 430000 402861 446261 369900 451950 359100 345000
## [31] 370878 350000 402000 423000 372500 392000 755000 361919 341000 538000
## [41] 395000 485000 582933 385000 350000 611657 395192 348000 556581 424870
## [51] 625000 392500 745000 367294 465000 378500 381000 410000 466500 377500
## [61] 394617
```

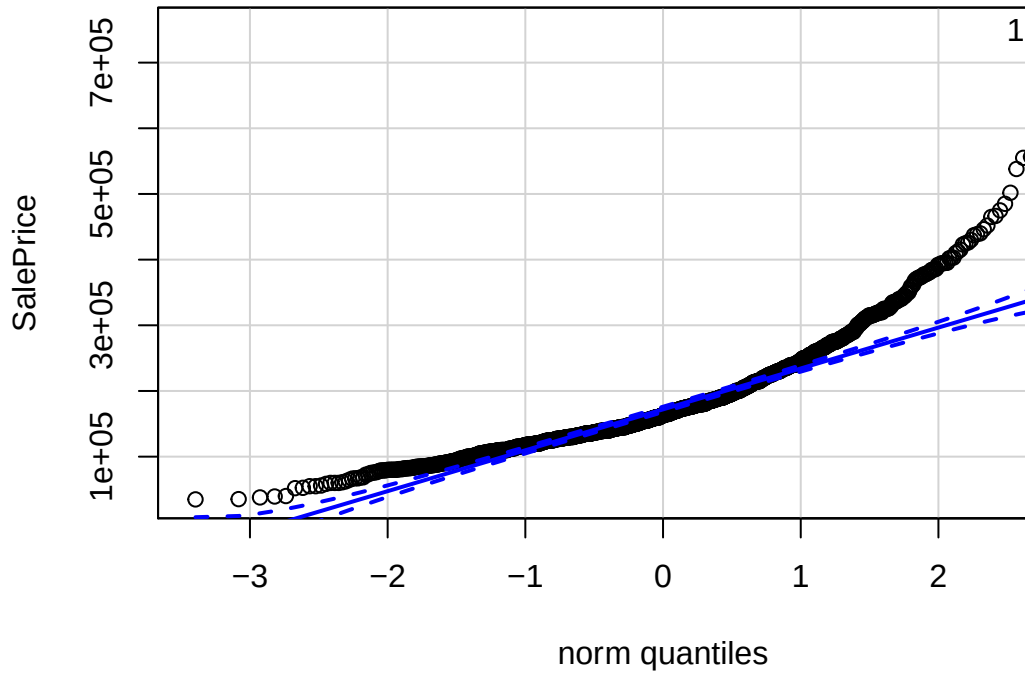
I want to remove outliers at least 3 standard deviations beyond the mean or higher, which is any value >=428240. Values >= 428249 are in rows 3, 7, 8, 17, 20, 21, 24, 26, 28, 37, 40, 42, 43, 46, 49, 51, 53, 55, 59. I remove these values manually.

```
## [1] 1441   80
```

Our summary statistics for the response variable have now changed due to the removal of Highly Suspect

Outliers. Our new mean is $176363, a decrease of 13,558 dollars or 17% of our old Standard Deviation. Our new median is $161750, a neglibible decrease 1,250 dollars or 0.8% of our original Median. Our new mean absolute deviation is $54,485.55 Our new Standard Deviation is $68,353.35 Our new skew is 1.04, an EXCELLENT IMPROVEMENT from 1.88 Our new kurtosis is 1.06, an EXCELLENT IMPROVEMENT from 6.5 This is starting to resemble a normal distribution!

```
##    vars    n     mean       sd median  trimmed      mad   min    max
## X1    1 1441 176362.5 68353.35 161750 169129.4 54485.55 34900 426000
##     range skew kurtosis      se
## X1 391100 1.04     1.06 1800.64
```



To prove it, let's check for normality.

```
## [1]  692 1183
```

```
##
##  Shapiro-Wilk normality test
##
## data:  SalePrice
## W = 0.86967, p-value < 2.2e-16
```

Let's try removing more outliers, say 2.5 standard deviations about the mean, that would be any value higher than $388,526.00. This has decreased our skew and kurtosis considerably, but it is still moderately positively skewed.

```
##    vars    n     mean       sd median  trimmed     mad   min    max  range
## X1    1 1426 173951.6 64506.97 160000 167875.3 53373.6 34900 386250 351350
##    skew kurtosis      se
## X1 0.88      0.6 1708.23
```

I'm going to remove one last set of outliers, this time ~1.5 standard deviations about the mean. This means

we will keep all data with a SalePrice less than about $300,000. This condition still preserves 92% of our original dataset while making our skew small enough to make the distribution approximately symmetric.

```
##    vars    n     mean       sd median  trimmed     mad   min    max  range
## X1    1 1345 164246.2 52094.67 156000 161634.3 48925.8 34900 299800 264900
##    skew kurtosis      se
## X1 0.42    -0.32 1420.47
```

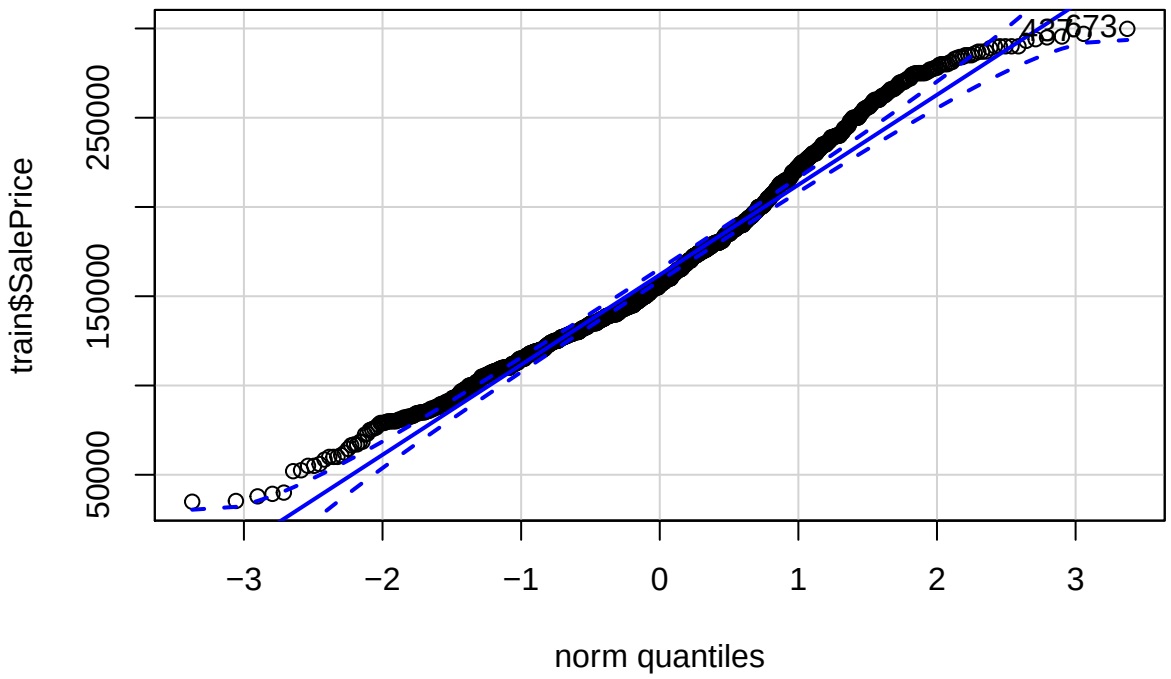
The values for asymmetry and kurtosis between -2 and +2 are considered acceptable in order to prove a normal univariate distribution (George & Mallery, 2010). George, D., & Mallery, M. (2010). SPSS for Windows Step by Step: A Simple Guide and Reference, 17.0 update (10a ed.) Boston: Pearson.

Skewness values between -0.5 and 0.5 are acceptable to assume an approximately symmetrical univariate distribution.

We satisfy both of these criteria, regardless of the Shapiro - Wilk test, so majority rules 2:1.

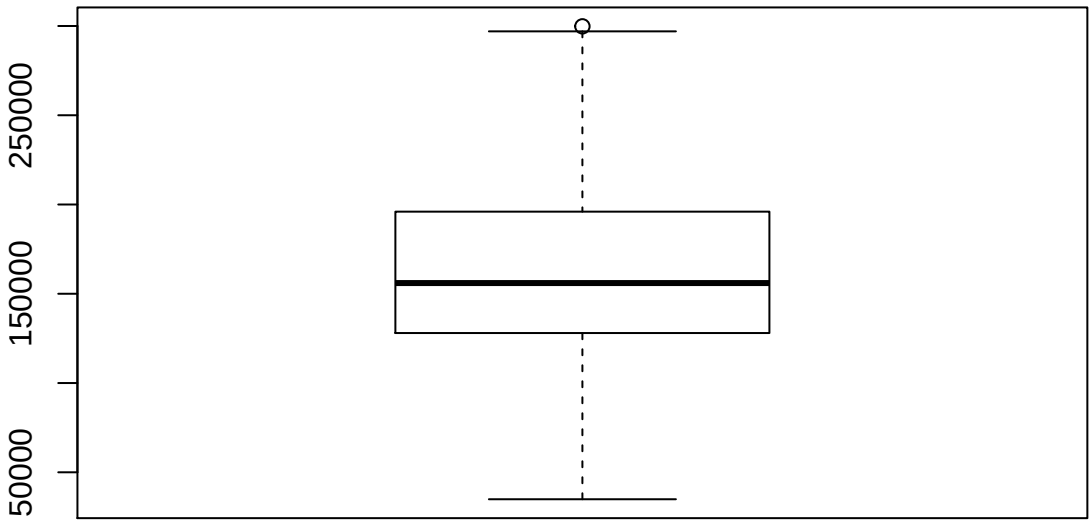
We can assume with reasonable confidence that the Response variable distribution is approximately normal.

The QQPlot is also much more linear and the boxplot is more even, with only one moderate outlier. I can live



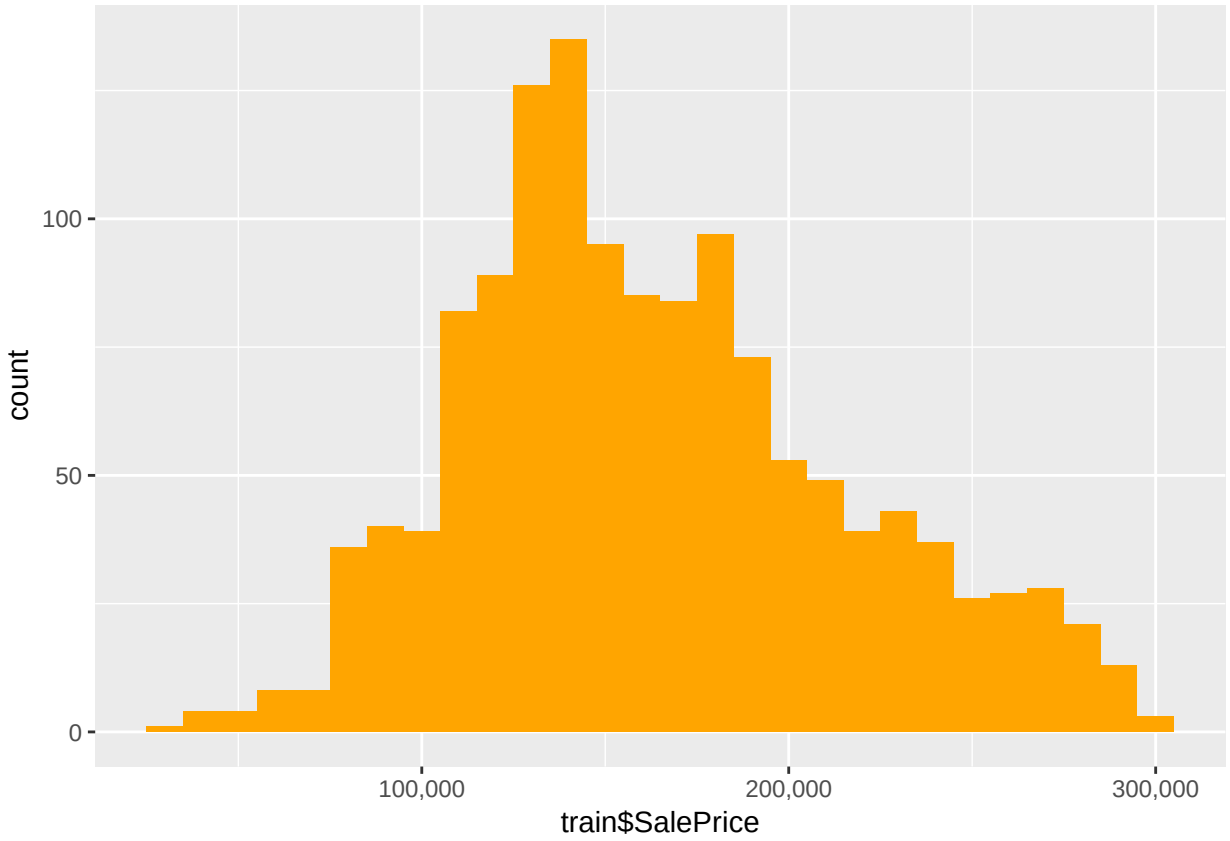
with that.

```
## [1] 673 437
```



As this dataset stands, we can now make normality assumptions due to the removal of skewness and kurtosis from the data. Since we have preserved about 92% of the original dataset we still have a sample space large enough to do meaningful inference on.

Before we decide which variables are important, let's plot our newly normalized SalePrice distribution



Let's figure out how much data is missing.

```
##                        Item Count
## MSSubClass       MSSubClass     0
## MSZoning           MSZoning     0
## LotFrontage     LotFrontage   250
## LotArea             LotArea     0
## Street               Street     0
## Alley                 Alley  1254
## LotShape           LotShape     0
## LandContour     LandContour     0
## Utilities         Utilities     0
## LotConfig         LotConfig     0
## LandSlope         LandSlope     0
## Neighborhood   Neighborhood     0
## Condition1       Condition1     0
## Condition2       Condition2     0
## BldgType           BldgType     0
## HouseStyle       HouseStyle     0
## OverallQual     OverallQual     0
## OverallCond     OverallCond     0
## YearBuilt         YearBuilt     0
## YearRemodAdd   YearRemodAdd     0
## RoofStyle         RoofStyle     0
## RoofMatl           RoofMatl     0
## Exterior1st     Exterior1st     0
## Exterior2nd     Exterior2nd     0
```

```
## MasVnrType       MasVnrType     7
## MasVnrArea       MasVnrArea     7
## ExterQual         ExterQual     0
## ExterCond         ExterCond     0
## Foundation       Foundation     0
## BsmtQual           BsmtQual    37
## BsmtCond           BsmtCond    37
## BsmtExposure   BsmtExposure    38
## BsmtFinType1   BsmtFinType1    37
## BsmtFinSF1       BsmtFinSF1     0
## BsmtFinType2   BsmtFinType2    38
## BsmtFinSF2       BsmtFinSF2     0
## BsmtUnfSF         BsmtUnfSF     0
## TotalBsmtSF     TotalBsmtSF     0
## Heating             Heating     0
## HeatingQC         HeatingQC     0
## CentralAir       CentralAir     0
## Electrical       Electrical     1
## X1stFlrSF         X1stFlrSF     0
## X2ndFlrSF         X2ndFlrSF     0
## LowQualFinSF   LowQualFinSF     0
## GrLivArea         GrLivArea     0
## BsmtFullBath   BsmtFullBath     0
## BsmtHalfBath   BsmtHalfBath     0
## FullBath           FullBath     0
## HalfBath           HalfBath     0
## BedroomAbvGr   BedroomAbvGr     0
## KitchenAbvGr   KitchenAbvGr     0
## KitchenQual     KitchenQual     0
## TotRmsAbvGrd   TotRmsAbvGrd     0
## Functional       Functional     0
## Fireplaces       Fireplaces     0
## FireplaceQu     FireplaceQu   686
## GarageType       GarageType    81
## GarageYrBlt     GarageYrBlt    81
## GarageFinish   GarageFinish    81
## GarageCars       GarageCars     0
## GarageArea       GarageArea     0
## GarageQual       GarageQual    81
## GarageCond       GarageCond    81
## PavedDrive       PavedDrive     0
## WoodDeckSF       WoodDeckSF     0
## OpenPorchSF     OpenPorchSF     0
## EnclosedPorch EnclosedPorch     0
## X3SsnPorch       X3SsnPorch     0
## ScreenPorch     ScreenPorch     0
## PoolArea           PoolArea     0
## PoolQC               PoolQC  1339
## Fence                 Fence  1069
## MiscFeature     MiscFeature  1291
## MiscVal             MiscVal     0
## MoSold               MoSold     0
## YrSold               YrSold     0
## SaleType           SaleType     0
```

```
## SaleCondition SaleCondition     0
## SalePrice         SalePrice     0
```

I'm gonna simplify things by just dropping the columns with too much missing data. The features I'm dropping are also very weak predictors.

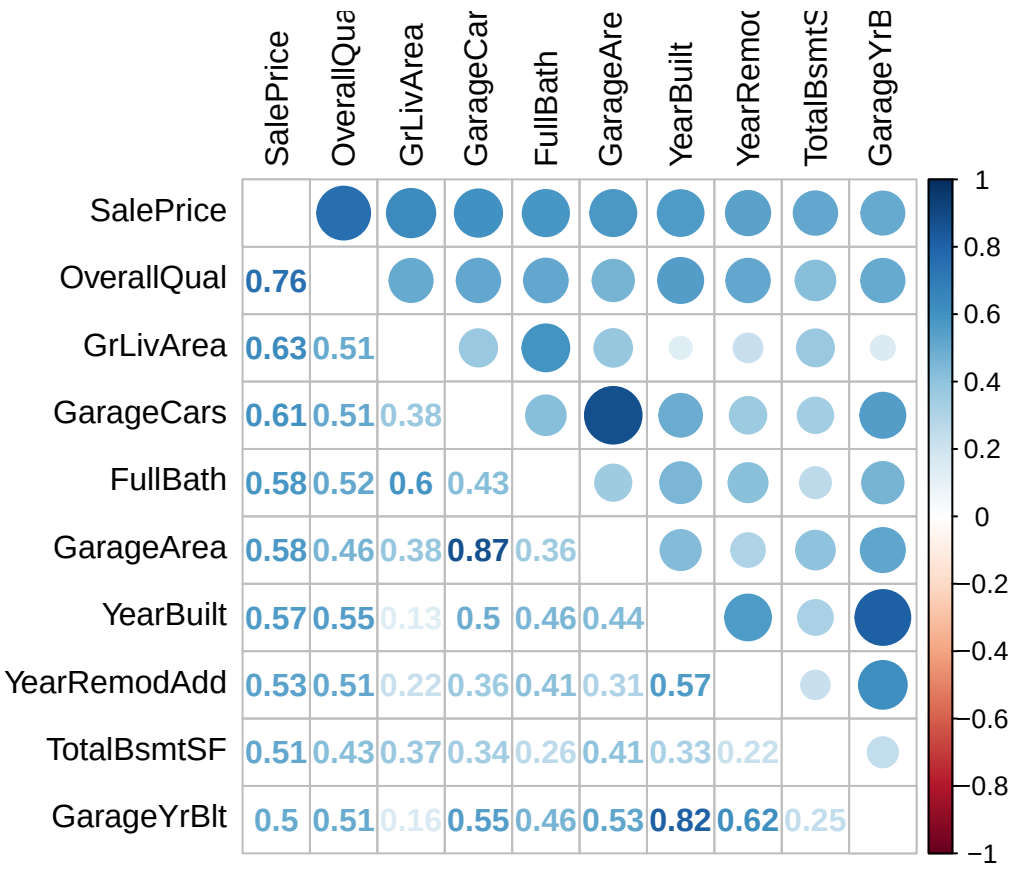
As we can see here, our top 8 most correlative variables with rho greater than 0.5 are listed in the first column of the correlation matrix.

All correlations are positive correlations, regardless of strength

1. Quality is the most Correlative variable. followed by:
2. Above ground living are
3. Garage car capacity
4. Number of full baths
5. Size of garage
6. Year built
7. Year Remodeled (if applicable)
8. Size of basement

However we do run into an issue with multicollinearity among features. This means we have to perform a Principal Component Analysis. We will kill two birds with one stone by performing a Principal Component Regression.

```
## We have 37 numeric variables and 38 categorical variables
```



Here I visualize all variables that have missing values and how many.

```
##  LotFrontage   GarageType  GarageYrBlt GarageFinish   GarageQual
##          250           81           81           81           81
```

```
##   GarageCond BsmtExposure BsmtFinType2     BsmtQual     BsmtCond
##           81           38           38           37           37
## BsmtFinType1   MasVnrType   MasVnrArea   Electrical
##           37            7            7            1
```

Since so few people can afford expensive houses, it makes sense to cater our model to the average buyer, by removing the skew of expensive outliers. This not only makes our response variable approximately normal and symmetrical, but we can know invoke some basic assumptions about the data, given that our variance is known, mainly that our data now abides by the central limit theorem stated below:

"if a population has finite variance sigma squared and a finite mean mu, then the distribution of sample means (from an infinite set of independent samples of N independent observations each) approaches a normal distribution (with variance ??2/N and mean ??) as the sample size increases, regardless of the shape of population distribution."

Next we apply a regression tree where OverallQuality, Ground Living Area, Neighborhood, Basement Size and GarageArea were the top variables used in prediction

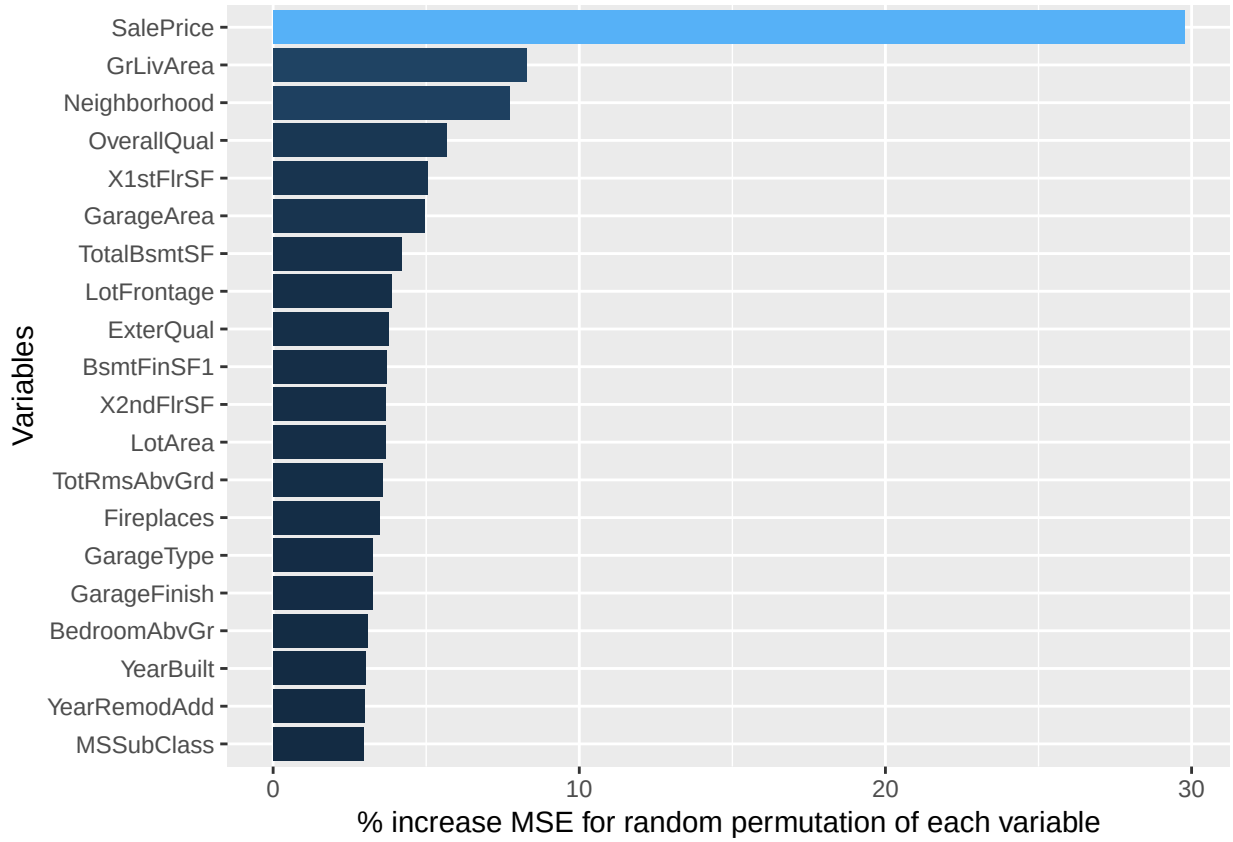
```
## [1] 1345   75
```

```
##
## Regression tree:
## tree(formula = train$SalePrice ~ ., data = train)
## Variables actually used in tree construction:
## [1] "OverallQual"  "GrLivArea"    "Neighborhood" "TotalBsmtSF"
## [5] "GarageArea"
## Number of terminal nodes:  11
## Residual mean deviance:  643200000 = 6.291e+11 / 978
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -105900.0  -14080.0    -106.5       0.0   14790.0  106400.0
```

```
## Warning in Ops.factor(left, right): '-' not meaningful for factors

## Warning in Ops.factor(left, right): '-' not meaningful for factors

## Warning in Ops.factor(left, right): '-' not meaningful for factors

## Warning in Ops.factor(left, right): '-' not meaningful for factors

## Warning in Ops.factor(left, right): '-' not meaningful for factors

## Warning in Ops.factor(left, right): '-' not meaningful for factors

## Warning in Ops.factor(left, right): '-' not meaningful for factors

## Warning in Ops.factor(left, right): '-' not meaningful for factors

## Warning in Ops.factor(left, right): '-' not meaningful for factors

## Warning in Ops.factor(left, right): '-' not meaningful for factors

## Warning in Ops.factor(left, right): '-' not meaningful for factors

## Warning in Ops.factor(left, right): '-' not meaningful for factors
```

```
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
## Warning in Ops.factor(left, right): '-' not meaningful for factors
```

```
## Warning in Ops.factor(left, right): '-' not meaningful for factors

## Warning in Ops.factor(left, right): '-' not meaningful for factors

## Warning in Ops.factor(left, right): '-' not meaningful for factors

## Warning in mean.default((yhat.boost - test)^2): argument is not numeric or
## logical: returning NA

## [1] NA

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   38226  127892  156211  164153  196857  304292

## [1] 1345   75

##      |      Out-of-bag   |
## Tree |      MSE  %Var(y) |
##  300 | 3.229e+07     1.19 |
##      |      Out-of-bag   |
## Tree |      MSE  %Var(y) |
##  300 | 3.044e+07     1.12 |
##      |      Out-of-bag   |
## Tree |      MSE  %Var(y) |
##  300 | 2.943e+07     1.09 |
##      |      Out-of-bag   |
## Tree |      MSE  %Var(y) |
##  300 | 3.402e+07     1.25 |
##      |      Out-of-bag   |
## Tree |      MSE  %Var(y) |
##  300 | 3.471e+07     1.28 |
```



This is enough data cleaning for me, since the most important variables are the most measured, we don't have to impute every single feature because most features that have weak importance aren't measured consistently across the dataset.

The basic idea behind PCR is to calculate the principal components and then use some of these components as predictors in a linear regression model fitted using the typical least squares procedure.

From Dr. Nguyen's notes:

"First we implement a LASSO and RIDGE Regression Model The `glmnet()` function has an `alpha` argument that determines what type of model is fit. If `alpha=0` then a ridge regression model is fit, and if `alpha=1` then a lasso model is fit. We first fit a ridge regression model. However, here we have chosen to implement the function over a grid of values ranging from $\lambda = 10^10$ to $\lambda = 10^{???2}$, essentially covering the full range of scenarios from the null model containing only the intercept, to the least squares fit."