Kaggle_Home_Prices_Ames_Iowa

```
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
library(scales)
library(dummies)
```

```
## dummies-1.5.6 provided by Decision Patterns
```

```
library(fmsb)
library(pls)
```

```
##
## Attaching package: 'pls'
```

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

```
library(randomForest)
```

```
## randomForest 4.6-12
```

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

```
##
## Attaching package: 'randomForest'
```

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

```
options(max.print=999999)
getwd()
```

```
## [1] "/Users/kevalshah/Google Drive/Research/Kaggle House_Prices"
```

```
train <- read.csv("train.csv", header = TRUE, sep = ",")
test <- read.csv("test.csv", header = TRUE, sep = ",")

# Add sale price new column in test dataset
test["SalePrice"] <- NA

# Let's explore the structure of the data
dim(train)
```

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

```
str(train)
```

```
## 'data.frame':    1460 obs. of  81 variables:
##  $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ MSSubClass   : int  60 20 60 70 60 50 20 60 50 190 ...
##  $ MSZoning     : 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 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 ...
```

The categorical variables are stored as factors in our dataframe.

```
# Combine Train and Test datasets
total <- rbind(train, test)

# Visualize missing data using ggplot and a function from neato package in R

library(reshape2)
library(ggplot2)
library(dplyr)
```

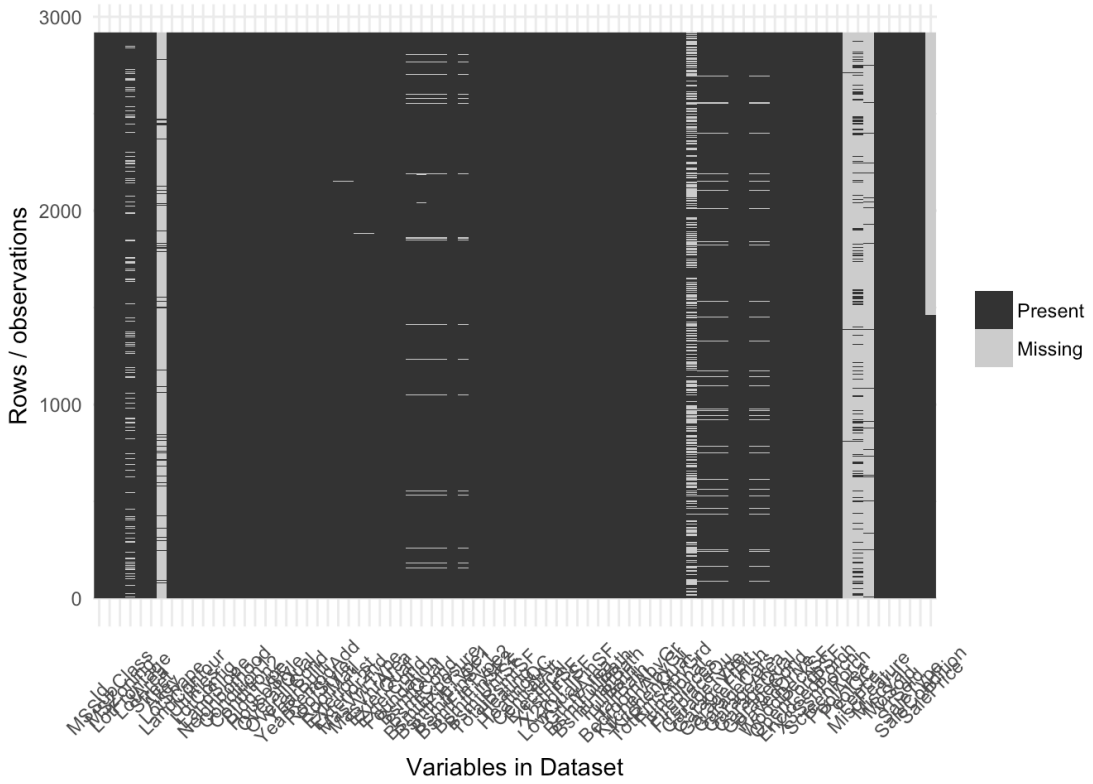
```
##
## Attaching package: 'dplyr'
```

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

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

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

```
ggplot_missing <- function(x){

   x %>%
    is.na %>%
    melt %>%
    ggplot(data = .,
           aes(x = Var2,
               y = Var1)) +
    geom_raster(aes(fill = value)) +
    scale_fill_grey(name = "",
                    labels = c("Present","Missing")) +
    theme_minimal() +
    theme(axis.text.x  = element_text(angle=45, vjust=0.5)) +
    labs(x = "Variables in Dataset",
         y = "Rows / observations")
}

ggplot_missing(total)
```




```
# Check for missing values
missing <- colSums(sapply(total, is.na))
missing
```

```
##            Id    MSSubClass      MSZoning   LotFrontage       LotArea
##             0             0             4           486             0
##        Street         Alley      LotShape   LandContour     Utilities
##             0          2721             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             1             1
##    MasVnrType    MasVnrArea     ExterQual     ExterCond    Foundation
##            24            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           157           159
##  GarageFinish    GarageCars    GarageArea    GarageQual    GarageCond
##           159             1             1           159           159
##    PavedDrive    WoodDeckSF   OpenPorchSF EnclosedPorch    X3SsnPorch
##             0             0             0             0             0
##   ScreenPorch      PoolArea        PoolQC         Fence   MiscFeature
##             0             0          2909          2348          2814
##       MiscVal        MoSold        YrSold      SaleType SaleCondition
##             0             0             0             1             0
##     SalePrice
##          1459
```

Data Cleaning Plan

Let’s look at each missing variables.

LotFrontage: 486 values missing. Linear feet of the street connected to property. Lot frontage, ideally, should #correlate with Lot Area. Also, check the lot shape and configuration of missing values.

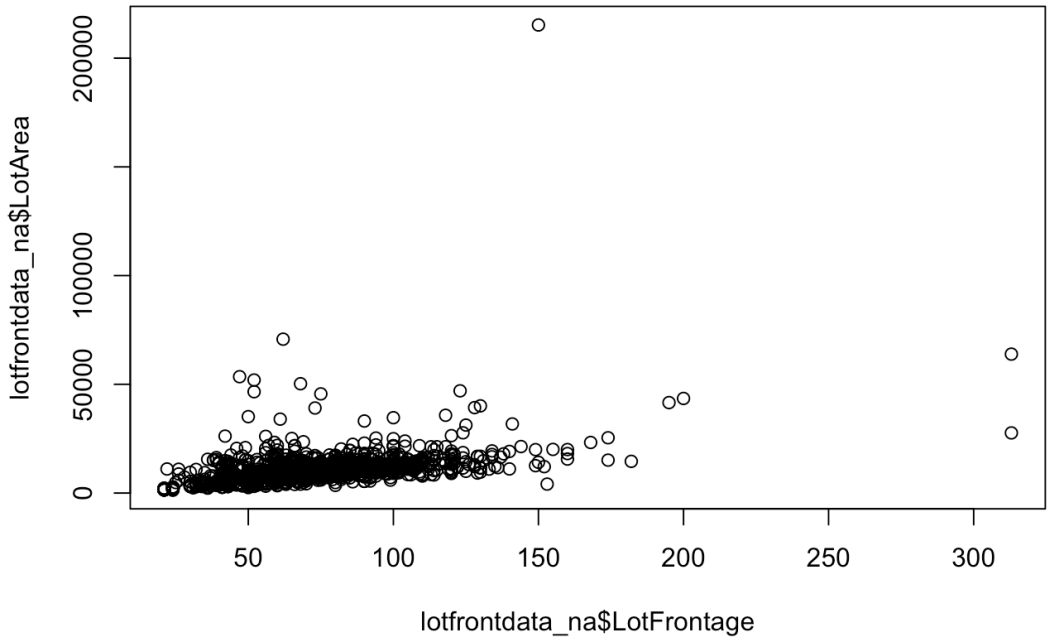
```
lotfront <- c("Id","LotFrontage","LotArea","LotShape","LotConfig")
lotfrontdata <- total[lotfront]
lotfrontdataNA <- lotfrontdata[is.na(lotfrontdata$LotFrontage),]
str(lotfrontdataNA)
```

```
## 'data.frame':    486 obs. of  5 variables:
##  $ Id         : int  8 13 15 17 25 32 43 44 51 65 ...
##  $ LotFrontage: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ LotArea    : int  10382 12968 10920 11241 8246 8544 9180 9200 13869 9375 ...
##  $ LotShape   : Factor w/ 4 levels "IR1","IR2","IR3",..: 1 2 1 1 1 1 1 1 2 4 ...
##  $ LotConfig  : Factor w/ 5 levels "Corner","CulDSac",..: 1 5 1 2 5 2 2 2 1 5 ...
```

```
#hist(lotfrontdataNA[c("Id","LotArea","LotShape","LotConfig")])
summary(lotfrontdataNA)
```

```
##        Id          LotFrontage     LotArea       LotShape    LotConfig
##  Min.   :   8.0   Min.   : NA   Min.   :  1533   IR1:321   Corner :104
##  1st Qu.: 721.2   1st Qu.: NA   1st Qu.:  8125   IR2: 28   CulDSac: 87
##  Median :1364.5   Median : NA   Median : 10452   IR3:  5   FR2    : 20
##  Mean   :1417.2   Mean   :NaN   Mean   : 12380   Reg:132   FR3    :  4
##  3rd Qu.:2142.8   3rd Qu.: NA   3rd Qu.: 12928             Inside :271
##  Max.   :2909.0   Max.   : NA   Max.   :164660
##                   NA's   :486
```

```
lotfrontdata_na <- na.omit(lotfrontdata)

plot(lotfrontdata_na$LotFrontage, lotfrontdata_na$LotArea)
```




```
# We take square root of LotArea to compute correlation with LotFrontage
cor(lotfrontdata_na$LotFrontage, sqrt(lotfrontdata_na$LotArea))
```

```
## [1] 0.647658
```

We see a slightly stronger correlation with Sq. root of Lot Area. However, the correlation is not very strong. We will substitute NAs for LotFrontage with mean value.

```
total$LotFrontage[is.na(total$LotFrontage)] <- round(mean(total$LotFrontage, na.rm = TRUE))
```

Categorical Missing Variables.

Some homes / properties do not have alley access.

```
total$Alley <- as.character(total$Alley)
total$Alley[is.na(total$Alley)] <- 'None'
total$Alley <- as.factor(total$Alley)
```

MasVnrType: Masonry veneer walls consist of a single non-structural external layer of masonry work, typically brick, backed by an air space. Here NA means that Masonry veneer wall is not existent.

MasVnrType and MasVnrArea have corresponding values of NA. Therefore, we set NA as None and MasVnrArea as 0.

```
total$MasVnrType <- as.character(total$MasVnrType)
total$MasVnrType[is.na(total$MasVnrType)] <- 'None'
total$MasVnrType <- as.factor(total$MasVnrType)

total$MasVnrArea <- as.numeric(total$MasVnrArea)
total$MasVnrArea[is.na(total$MasVnrArea)] <- '0'
total$MasVnrArea <- as.numeric(total$MasVnrArea)
```

According to data dictionary, BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, BsmtFinType2, FireplaceQU = NA means that the properties or homes do not have a basement.

```
total$BsmtQual <- as.character(total$BsmtQual)
total$BsmtQual[is.na(total$BsmtQual)] <- 'None'
total$BsmtQual <- as.factor(total$BsmtQual)
```

```
total$BsmtCond <- as.character(total$BsmtCond)
total$BsmtCond[is.na(total$BsmtCond)] <- 'None'
total$BsmtCond <- as.factor(total$BsmtCond)
```

```
total$BsmtExposure <- as.character(total$BsmtExposure)
total$BsmtExposure[is.na(total$BsmtExposure)] <- 'None'
total$BsmtExposure <- as.factor(total$BsmtExposure)
```

```
total$BsmtFinType1 <- as.character(total$BsmtFinType1)
total$BsmtFinType1[is.na(total$BsmtFinType1)] <- 'None'
total$BsmtFinType1 <- as.factor(total$BsmtFinType1)
```

```
total$BsmtFinType2 <- as.character(total$BsmtFinType2)
total$BsmtFinType2[is.na(total$BsmtFinType2)] <- 'None'
total$BsmtFinType2 <- as.factor(total$BsmtFinType2)
```

```
total$Electrical <- as.character(total$Electrical)
total$Electrical[is.na(total$Electrical)] <- 'None'
total$Electrical <- as.factor(total$Electrical)
```

```
total$FireplaceQu <- as.character(total$FireplaceQu)
total$FireplaceQu[is.na(total$FireplaceQu)] <- 'None'
total$FireplaceQu <- as.factor(total$FireplaceQu)
```

```
total$GarageType <- as.character(total$GarageType )
total$GarageType[is.na(total$GarageType )] <- 'None'
total$GarageType <- as.factor(total$GarageType)
```

```
total$GarageYrBlt <- as.numeric(total$GarageYrBlt )
total$GarageYrBlt[is.na(total$GarageYrBlt )] <- '0'
total$GarageYrBlt <- as.numeric(total$GarageYrBlt)
```

```
total$GarageFinish <- as.character(total$GarageFinish )
total$GarageFinish[is.na(total$GarageFinish )] <- 'None'
total$GarageFinish <- as.factor(total$GarageFinish)
```

```
total$GarageQual <- as.character(total$GarageQual )
total$GarageQual[is.na(total$GarageQual )] <- 'None'
total$GarageQual <- as.factor(total$GarageQual)
```

```
total$GarageCond <- as.character(total$GarageCond )
total$GarageCond[is.na(total$GarageCond )] <- 'None'
total$GarageCond <- as.factor(total$GarageCond)
```

```
total$PoolQC <- as.character(total$PoolQC )
total$PoolQC[is.na(total$PoolQC )] <- 'None'
total$PoolQC <- as.factor(total$PoolQC)
```

```
total$Fence <- as.character(total$Fence )
total$Fence[is.na(total$Fence )] <- 'None'
total$Fence <- as.factor(total$Fence)
```

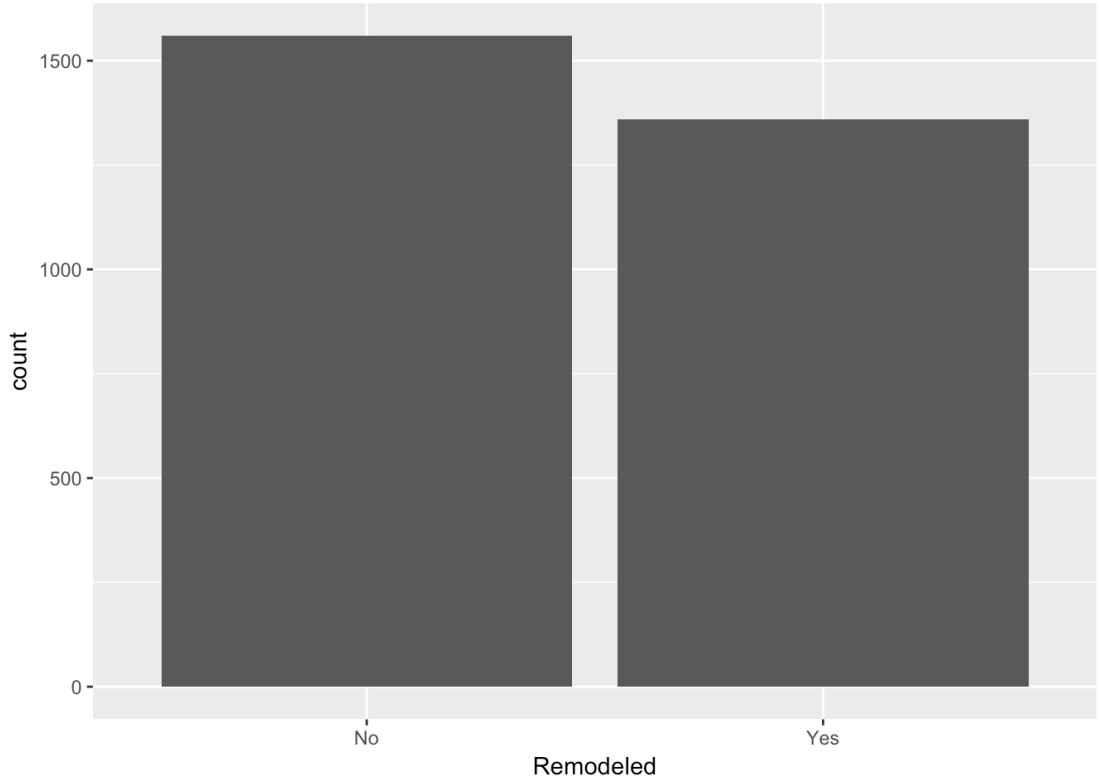
```
total$MiscFeature <- as.character(total$MiscFeature )
total$MiscFeature[is.na(total$MiscFeature )] <- 'None'
total$MiscFeature <- as.factor(total$MiscFeature)
```

All missing values have either been imputed or filled with more meaningful values.

Let’s explore the variable year built and year remodeled. The data dictionary states that if the year built is different from year remodeled, then the house

was remodeled. We will create another column, a binary value/flag for remodeled.

```
total$Remodel_flag <- "Yes"
total[total$YearBuilt==total$YearRemodAdd,]$Remodel_flag <- "No"
total$Remodel_flag <- as.factor(total$Remodel_flag)
# Number of remodeled homes
ggplot(total, aes(x = factor(total$Remodel_flag))) + geom_bar(stat = "count") + xlab("Remodeled")
```




```
# Percentage of remodeled homes
paste(round(sum(total$Remodel_flag == "Yes")/nrow(total)*100, 2), '%')
```

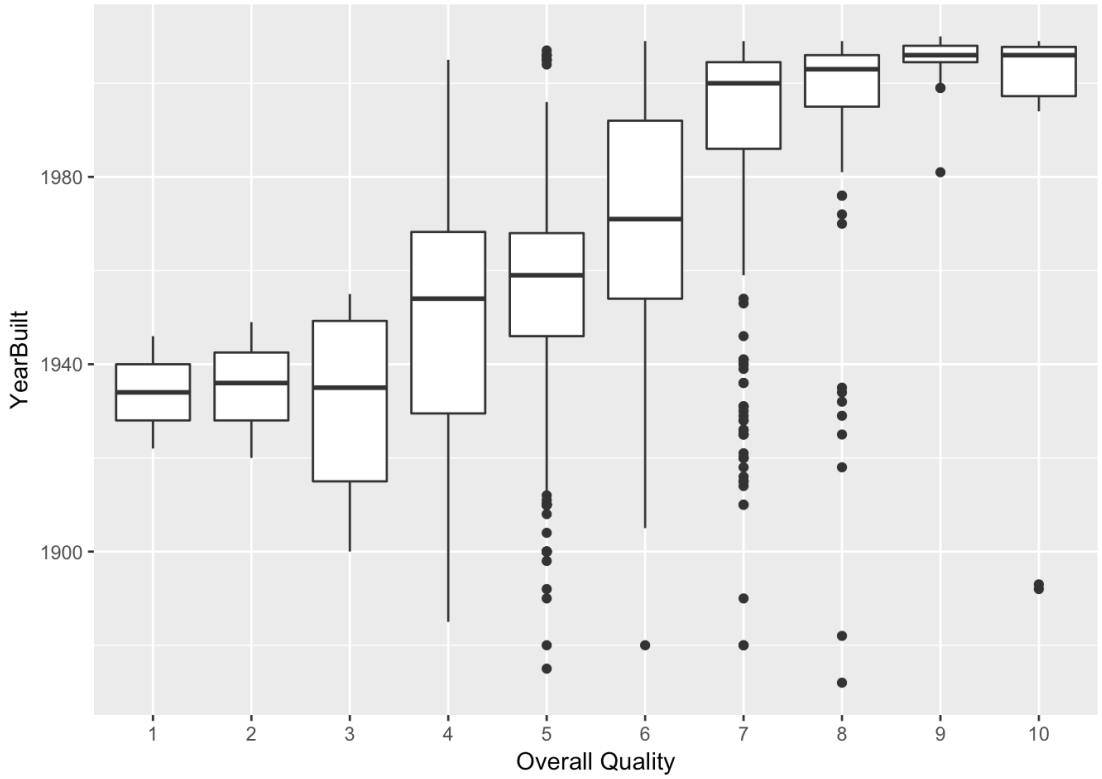
```
## [1] "46.56 %"
```

Split data into train and test

```
train <- total[1:1460,]
test <- total[1461:2919,]
```

Exploratory data analysis plan

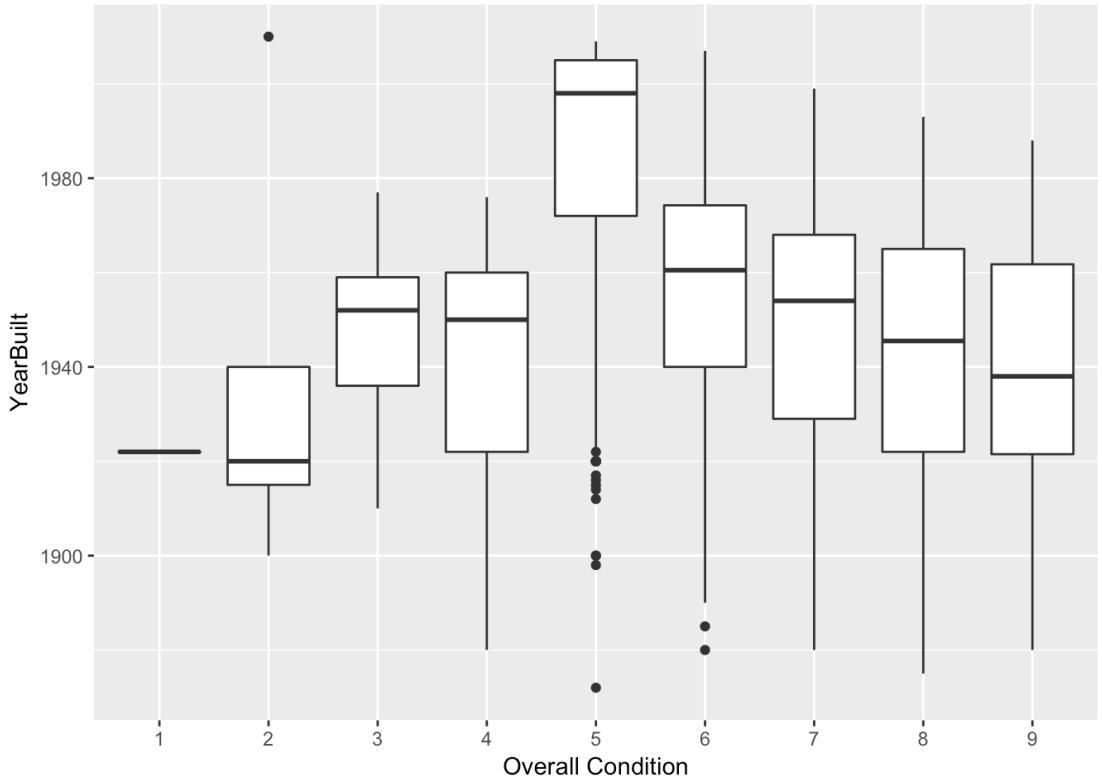
```
ggplot(train, aes(factor(OverallQual),YearBuilt)) + geom_boxplot() +xlab("Overall Quality")
```



#Most recently built homes

have better overall quality. Overall quality rates the material and finish of homes.

```
ggplot(train, aes(factor(OverallCond),YearBuilt)) + geom_boxplot() +xlab("Overall Condition")
```



#The recently built homes

have better Qverall Quality, but the Overall condition of these recently built homes is worse than the old homes. Newer built homes are of mediocre quality.

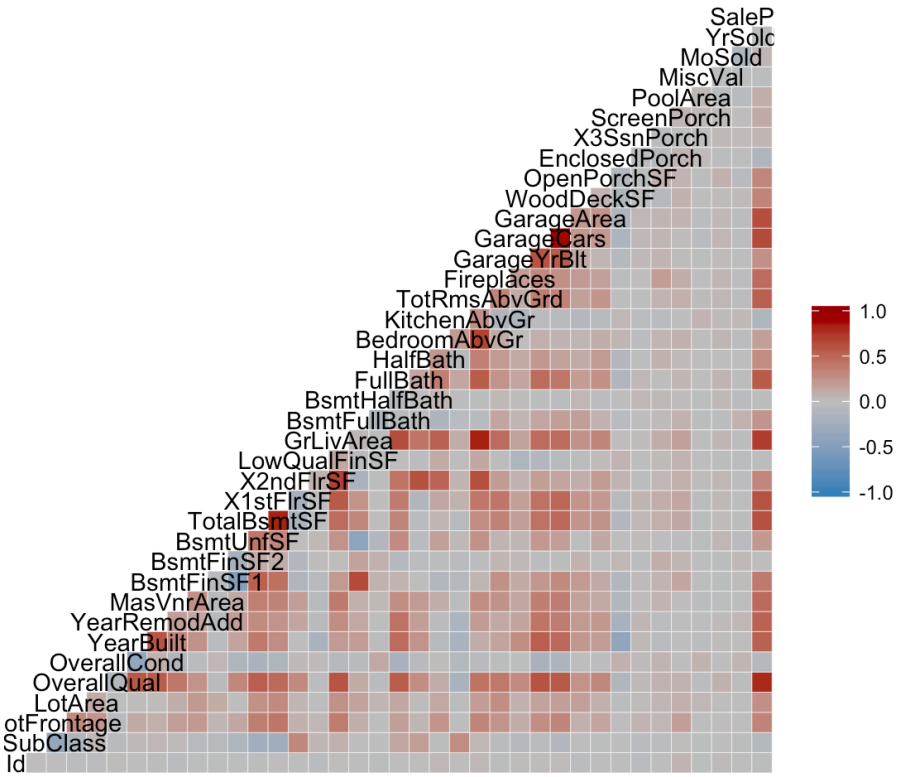
Let’s plot the correlation matrix of numeric variables in the dataset

```
train_num <- train[sapply(train,is.numeric)]

#correlations <- cor(na.omit(train_num))
#row_indic <- apply(correlations, 1, function(x) sum(x > 0.3 | x < -0.3) > 1)
#correlations<- correlations[row_indic ,row_indic ]
#corrplot(correlations, method="square")

# Another way to visualize correlation matrix
library(GGally)
```

```
##
## Attaching package: 'GGally'
```

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

```
ggcorr(train_num, low = "steelblue", mid = "grey", high = "darkred")
```




```
correlations <- cor(na.omit(train_num[2:37,]))
```

```
## Warning in cor(na.omit(train_num[2:37, ])): the standard deviation is zero
```

Let's make some scatter plot for some of the high correlation variables. High correlation variables:

OverallQual: Rates the overall material and finish of the house 1-10.

YearBuilt: Year house was built

MasVnrArea: Masonary veener area in square feet

TotalBsmtSF: Total Square feet of basement Area

X1stFlrSF: First floor Square feet

GrLivArea: Ground Living Area

FullBath: Full Bathrooms above grade

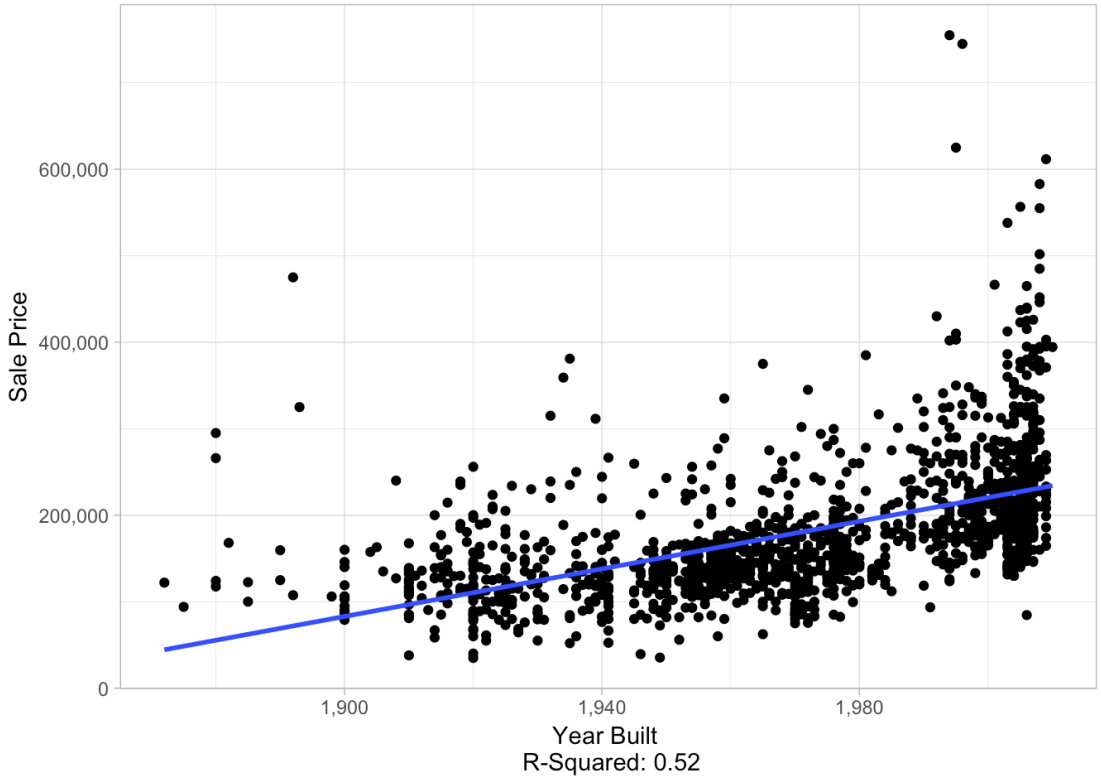
TotRmsAbvGrd: Total rooms above grade (doesn't include bathrooms)

GarageCars: Size of garage in car capacity

GarageArea: Size of garage in square feet

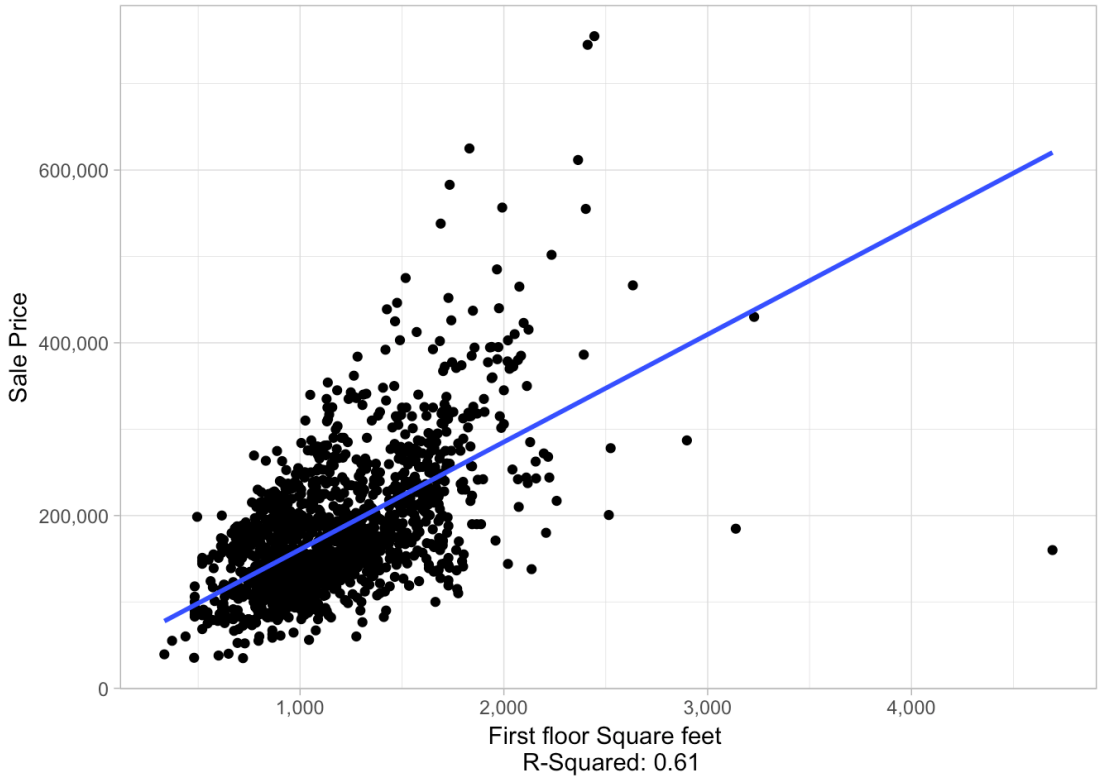
```
makeScatterplots <- function(dataframe,x.variable, y.variable, xlabel, ylabel){
  p = ggplot(dataframe, aes_string(x=x.variable,y= y.variable)) + geom_point() + geom_smooth(method=lm, se=F
ALSE) + ylab(ylabel) + xlab(paste(xlabel,'\n', 'R-Squared:', round(cor(x.variable, y.variable), 2))) + theme
_light() + scale_x_continuous(labels = comma) + scale_y_continuous(labels = comma)
  return(p)
}
```

```
makeScatterplots(train_num, train_num$YearBuilt, train_num$SalePrice, "Year Built", "Sale Price")
```



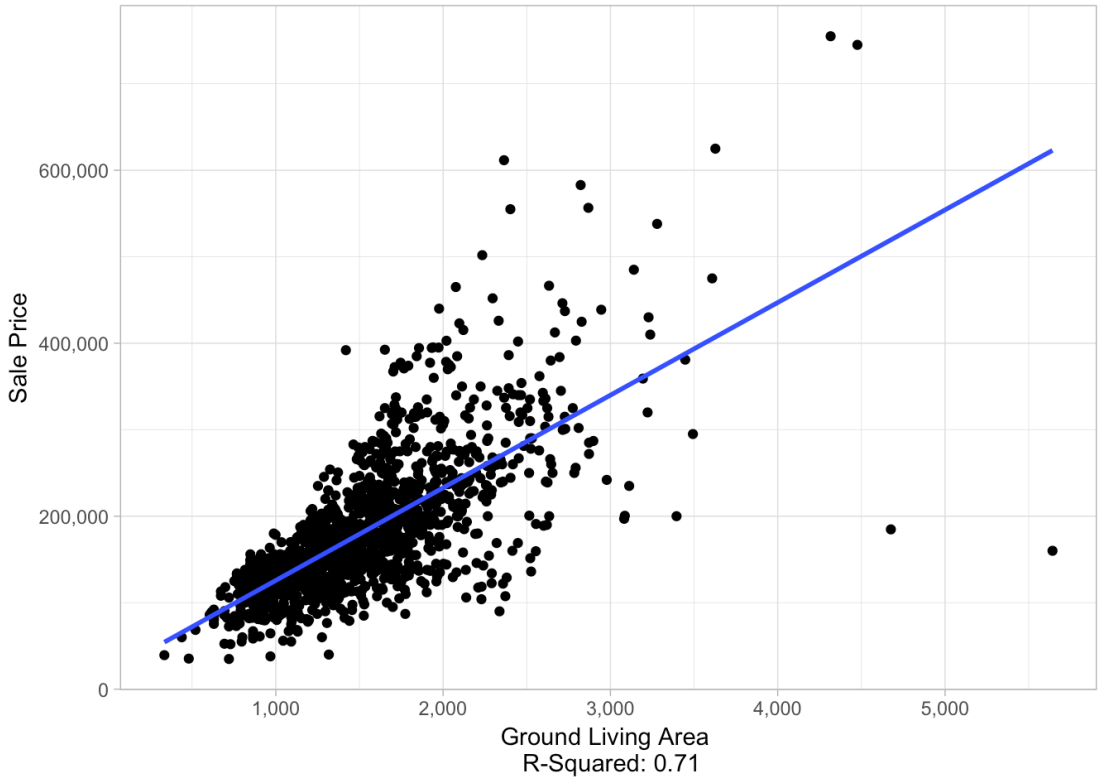
#There are a few recently built homes that are outliers and have much higher sale price.

```
makeScatterplots(train_num, train_num$X1stFlrSF, train_num$SalePrice, "First floor Square feet", "Sale Pric
e")
```



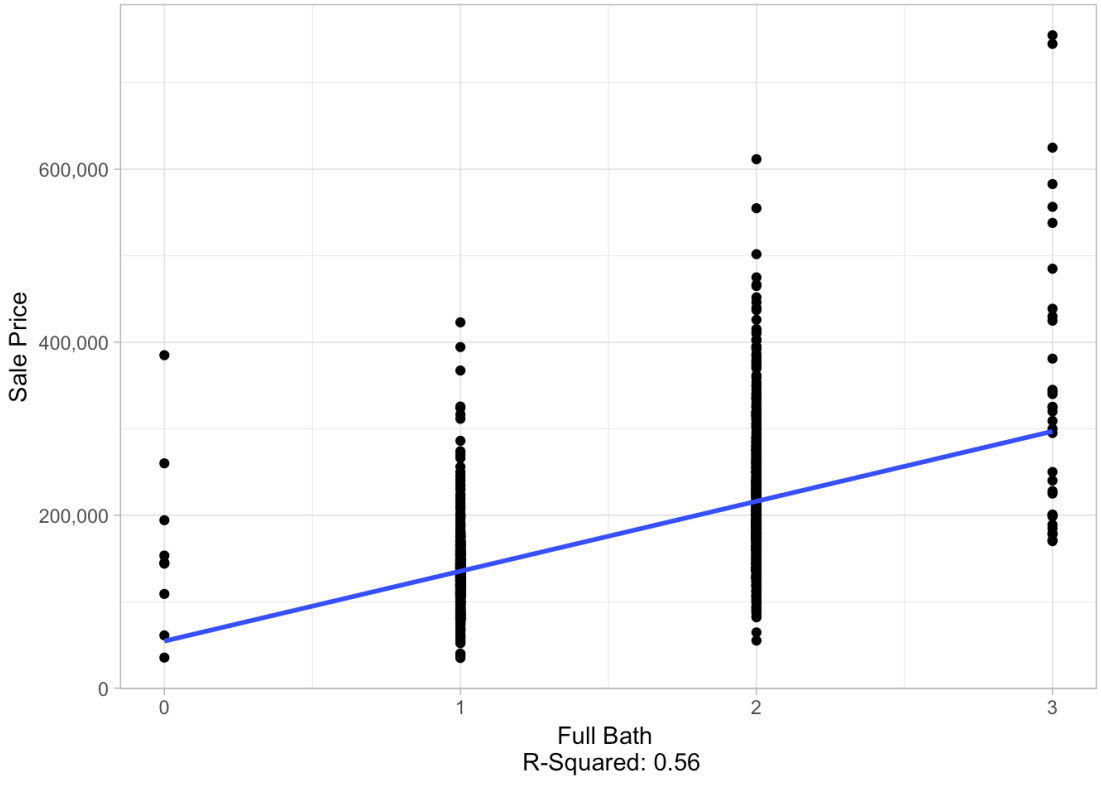
A majority of homes are under $200,000 and Average square foot of first floor in homes is ~1163 sq.feet.

```
makeScatterplots(train_num, train_num$GrLivArea, train_num$SalePrice, "Ground Living Area", "Sale Price")
```

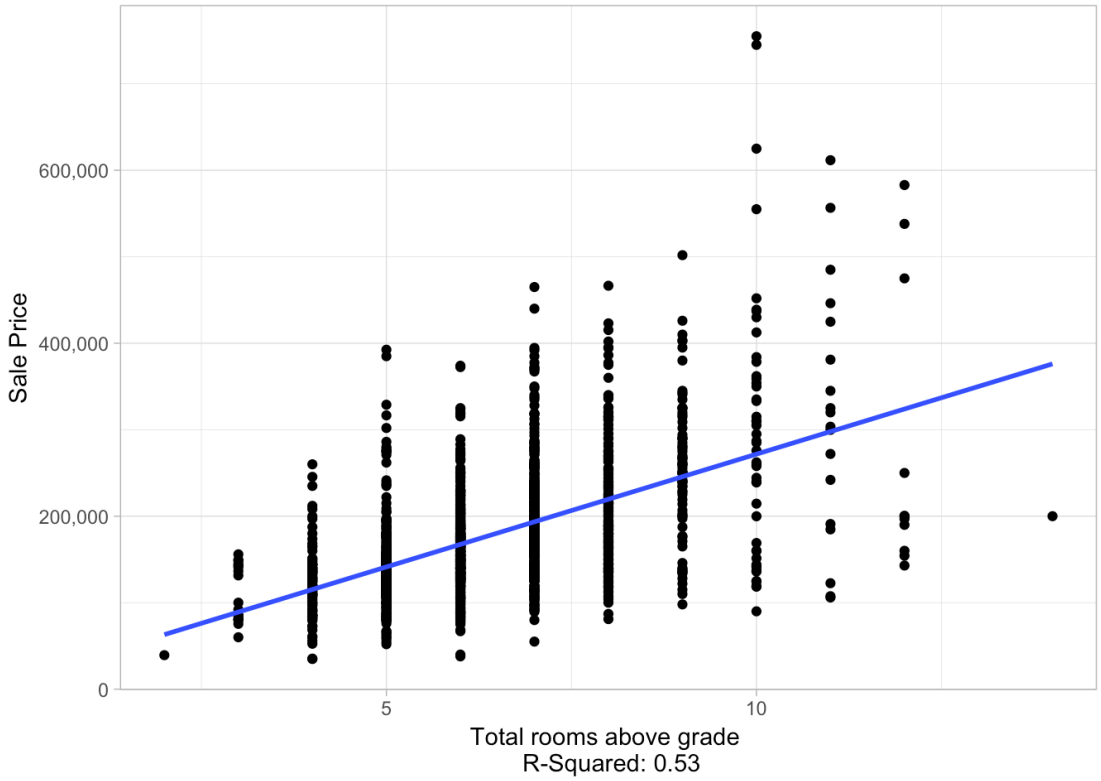


Living areas tend to be around an average of 1500 sq.feet for most homes.

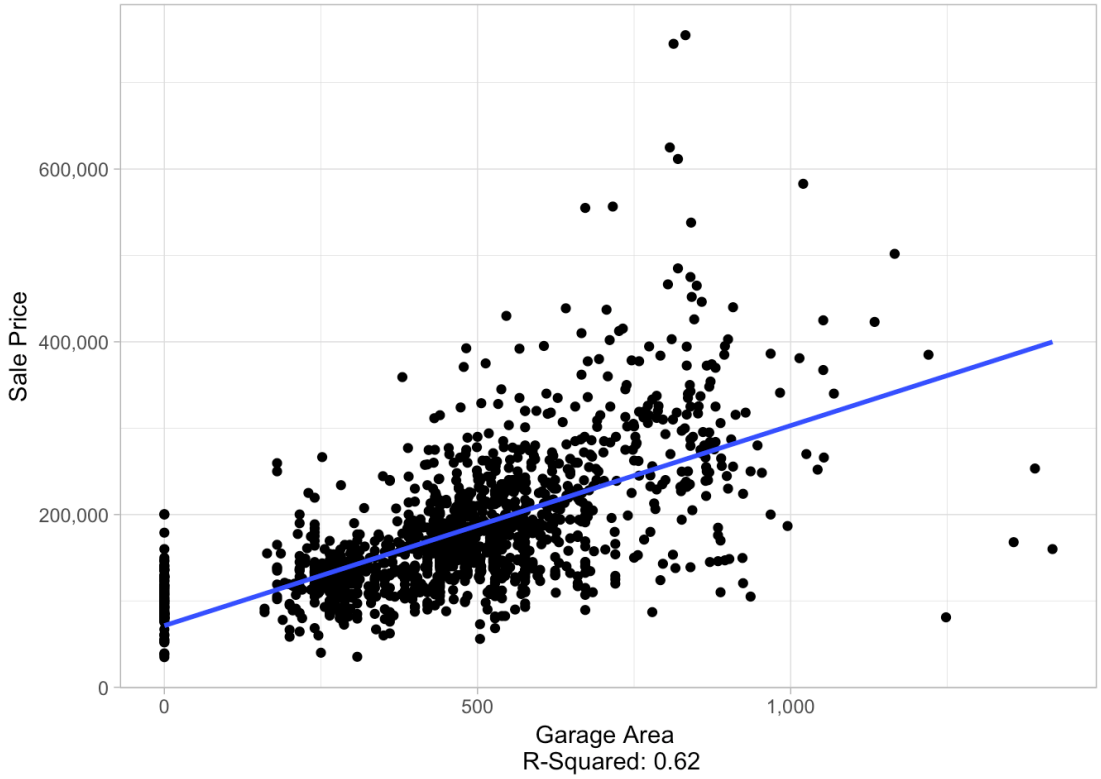
```
makeScatterplots(train_num, train_num$FullBath, train_num$SalePrice, "Full Bath", "Sale Price")
```



```
makeScatterplots(train_num, train_num$TotRmsAbvGrd, train_num$SalePrice, "Total rooms above grade", "Sale Pr
ice")
```

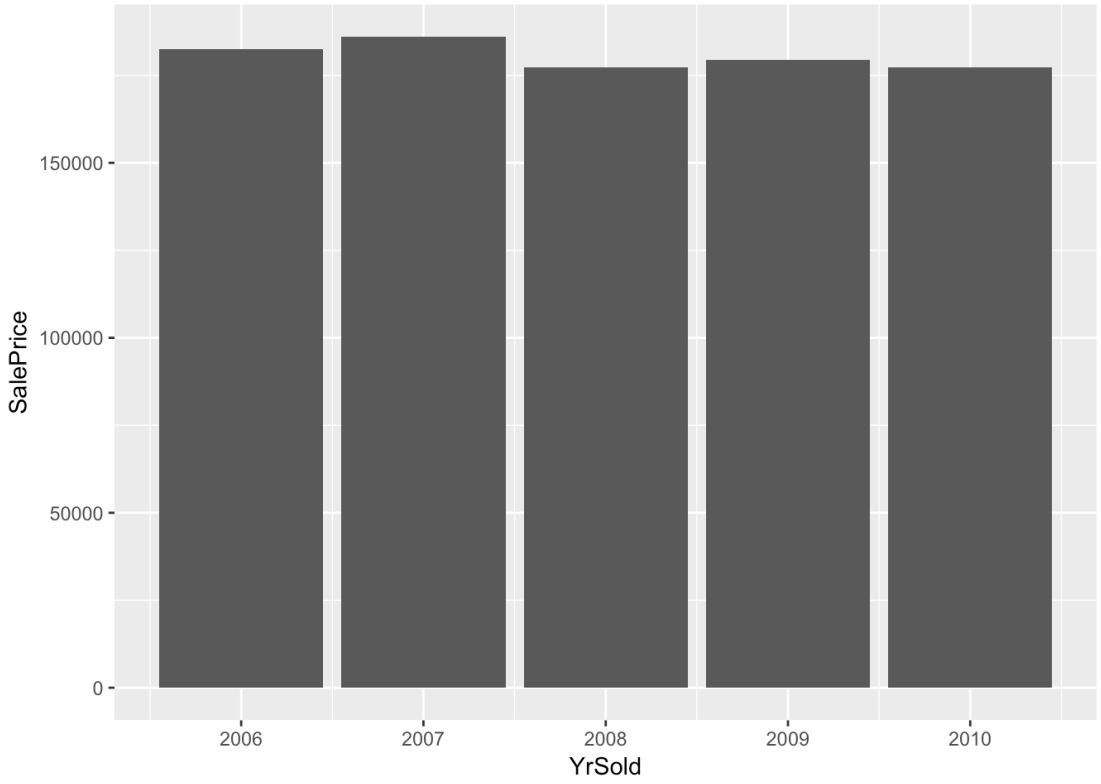



```
makeScatterplots(train_num, train_num$GarageArea, train_num$SalePrice, "Garage Area", "Sale Price")
```



#The sale prices of home is higher for garage areas between 750 to 1000 sq.feet. However, there are a few outliers where sale price drops for homes where garage area is greater than ~1000 sq.feet.

```
ggplot(train, aes(x=YrSold, y=SalePrice)) + stat_summary(fun.y="mean", geom="bar")
```



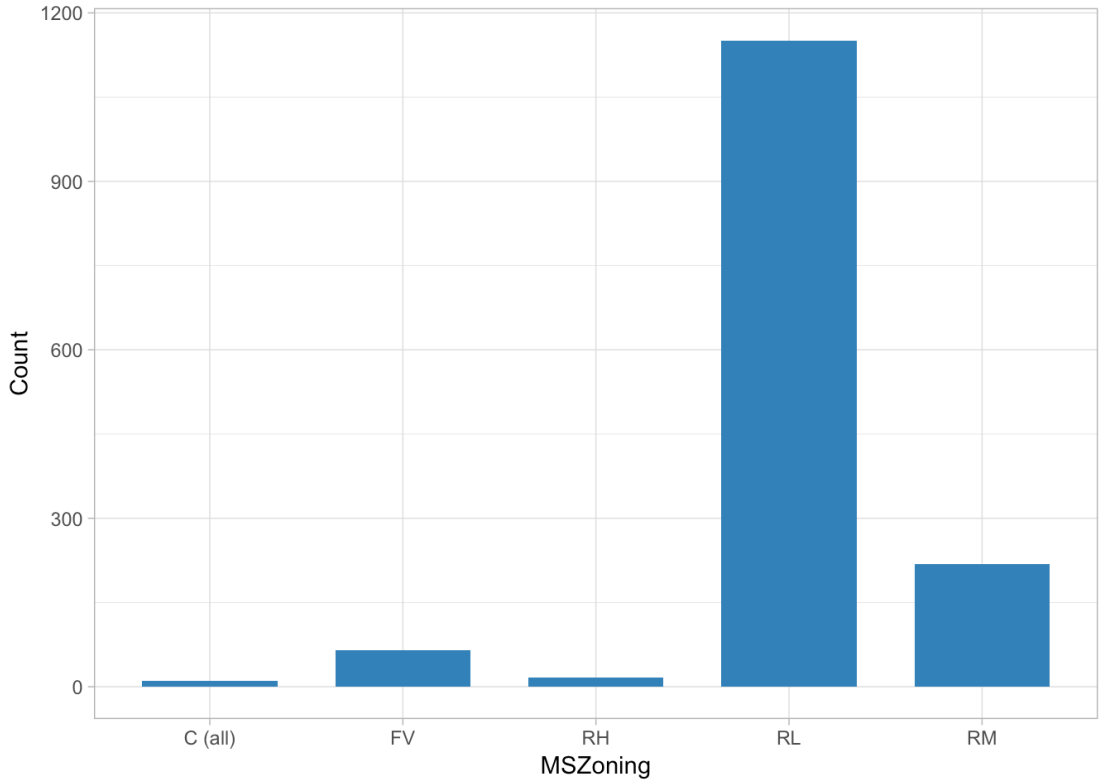
Notice the drop in average sale of home price in year 2008, the housing market bubble crashed when Case-Shiller home price index reported it's largest price drop.

Categorical Variables

Let's make some barplots for categorical variables to get a deeper insight / understanding of our data.

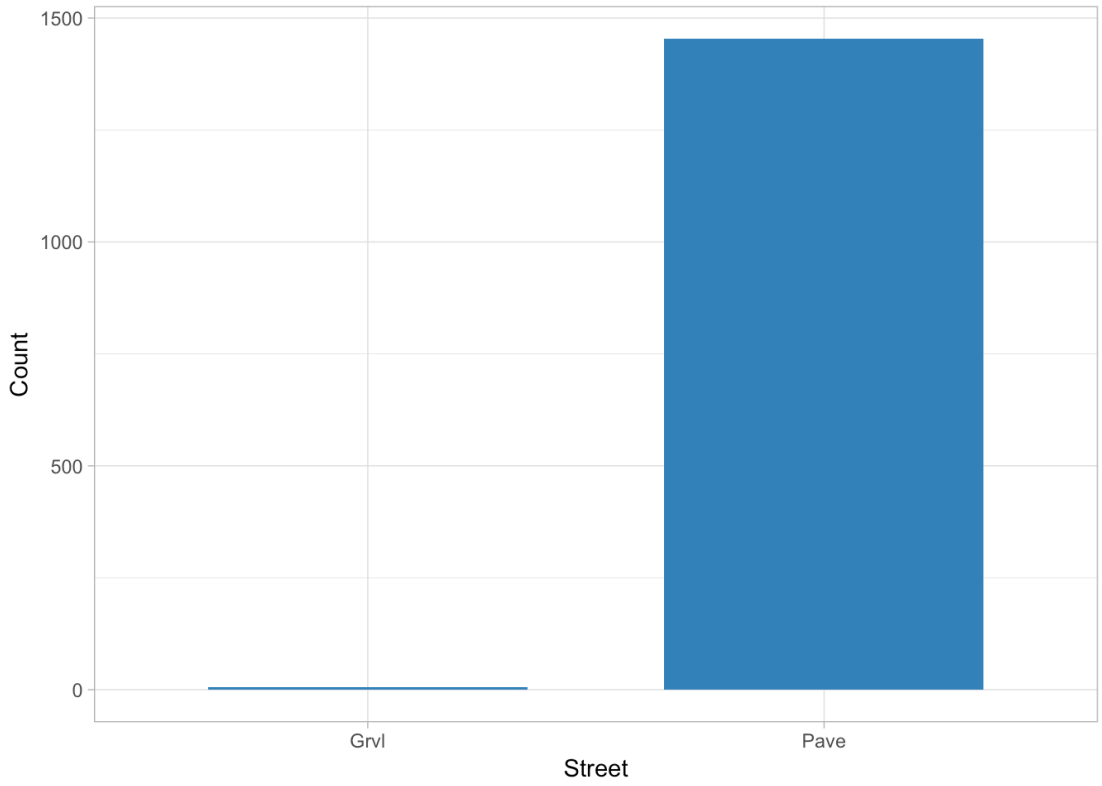
```
makeBarplots <- function(dataframe,x.variable, xlabel, ylabel){
  p = ggplot(dataframe, aes(x=factor(x.variable))) + geom_bar(stat = "count", width=0.7, fill="steelblue") +
 ylab(ylabel) + xlab(xlabel) + theme_light()
  return(p)
}
```

```
makeBarplots(train, train$MSZoning, "MSZoning", "Count")
```

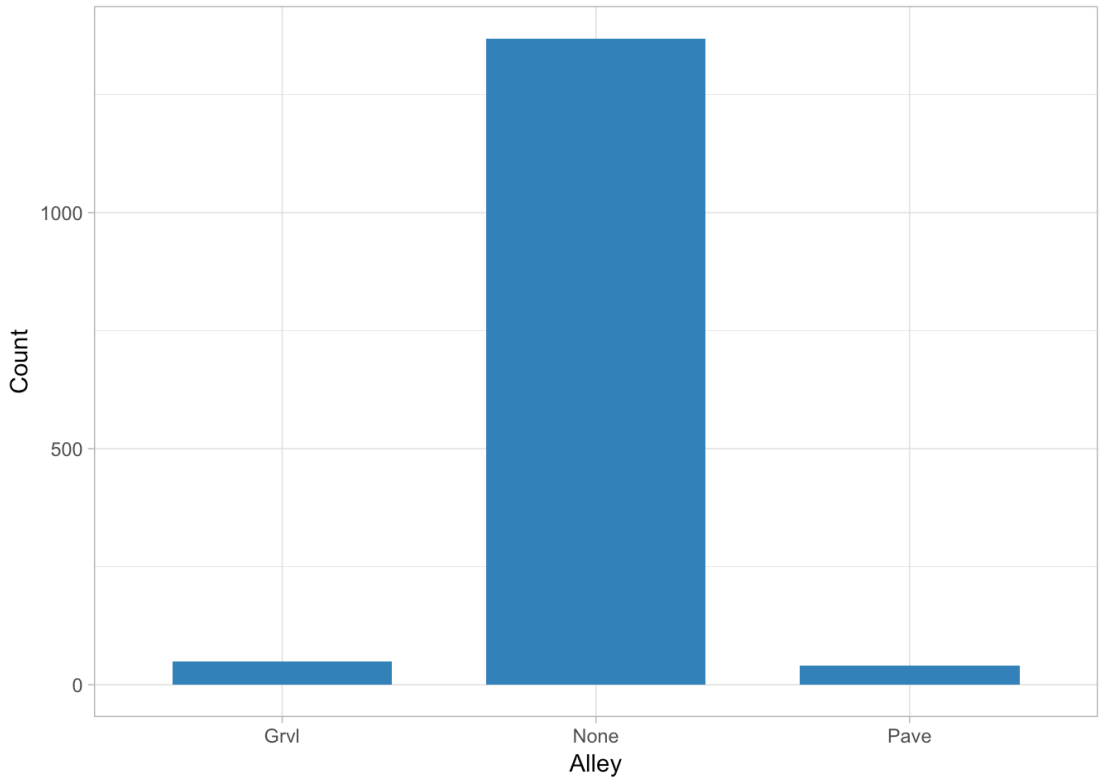


#An overwhelming majority of homes are in Residential Low Density zone.

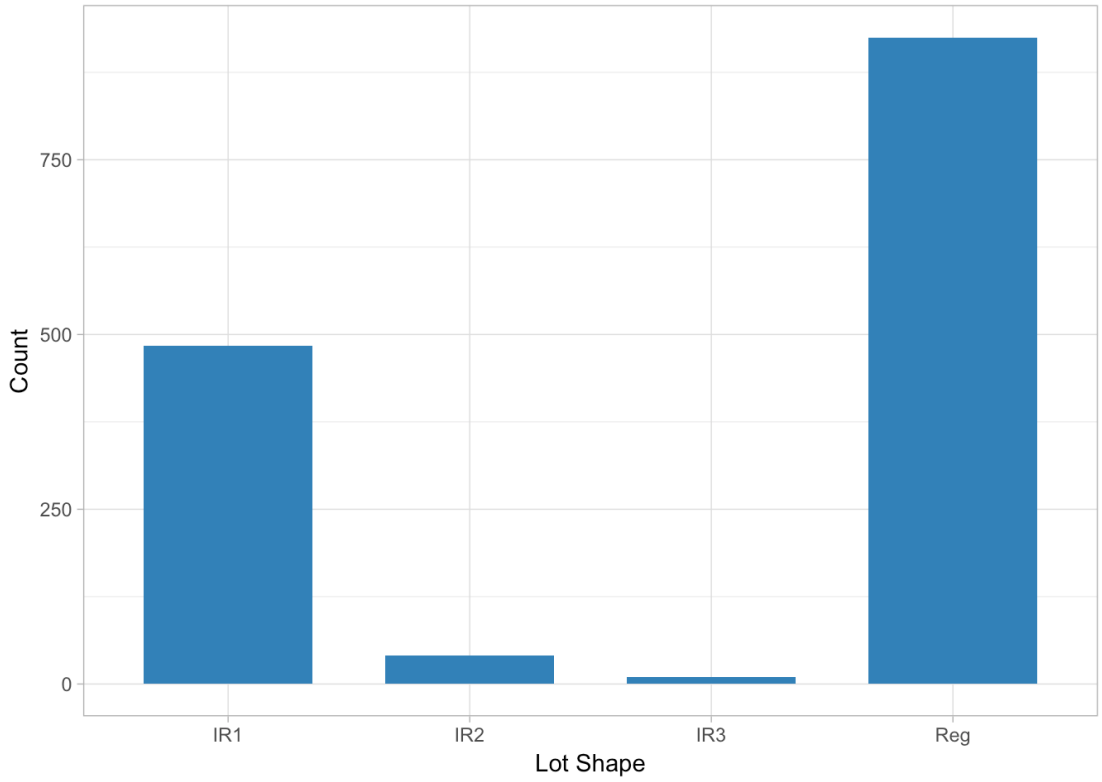
```
makeBarplots(train, train$Street, "Street", "Count")
```



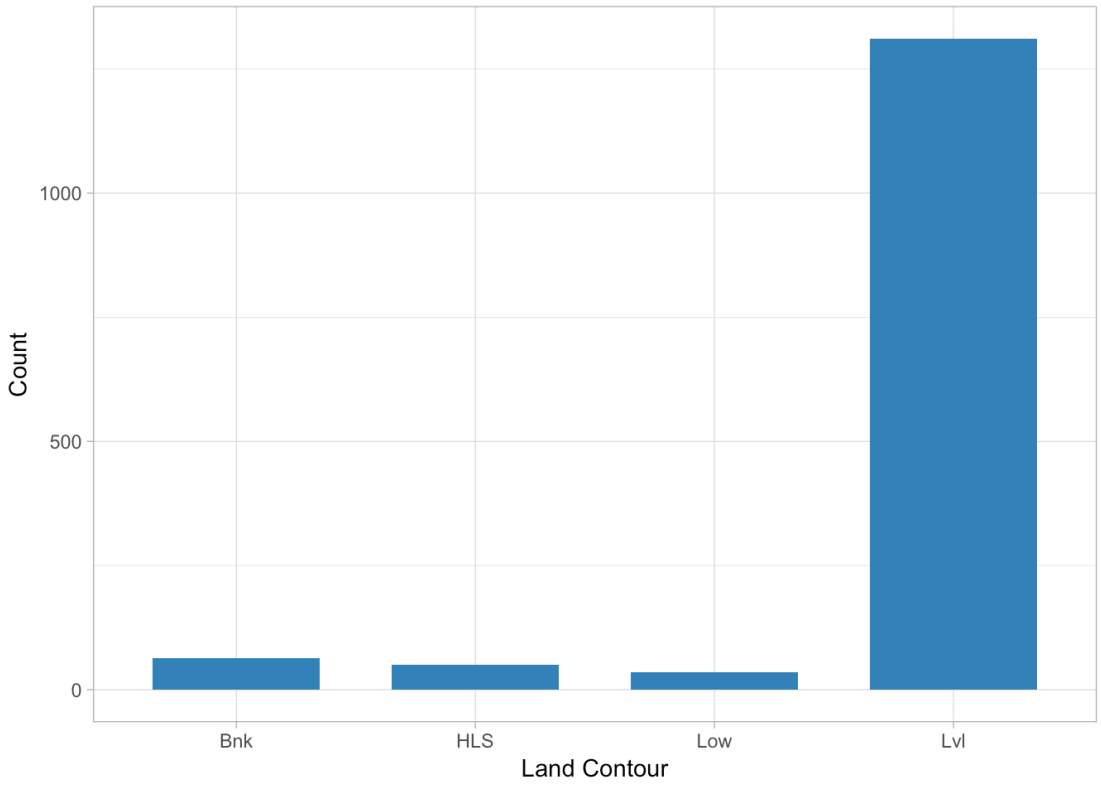

```
makeBarplots(train, train$Alley, "Alley", "Count")
```



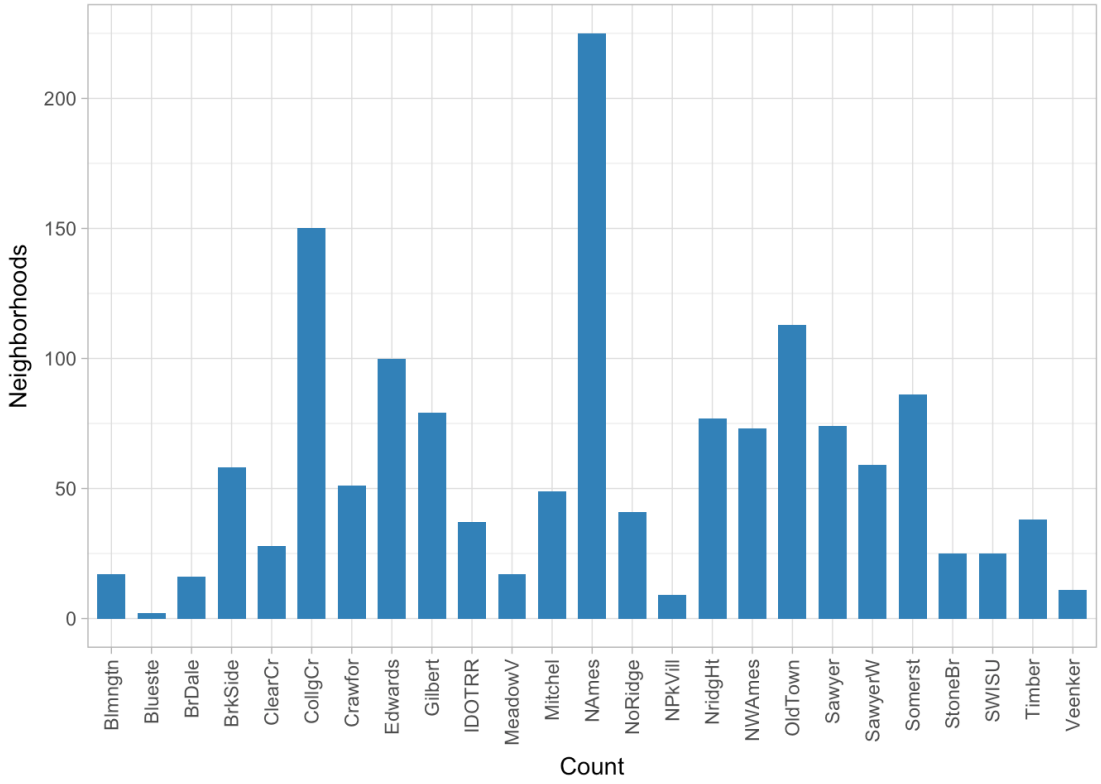

```
makeBarplots(train, train$LotShape, "Lot Shape", "Count")
```



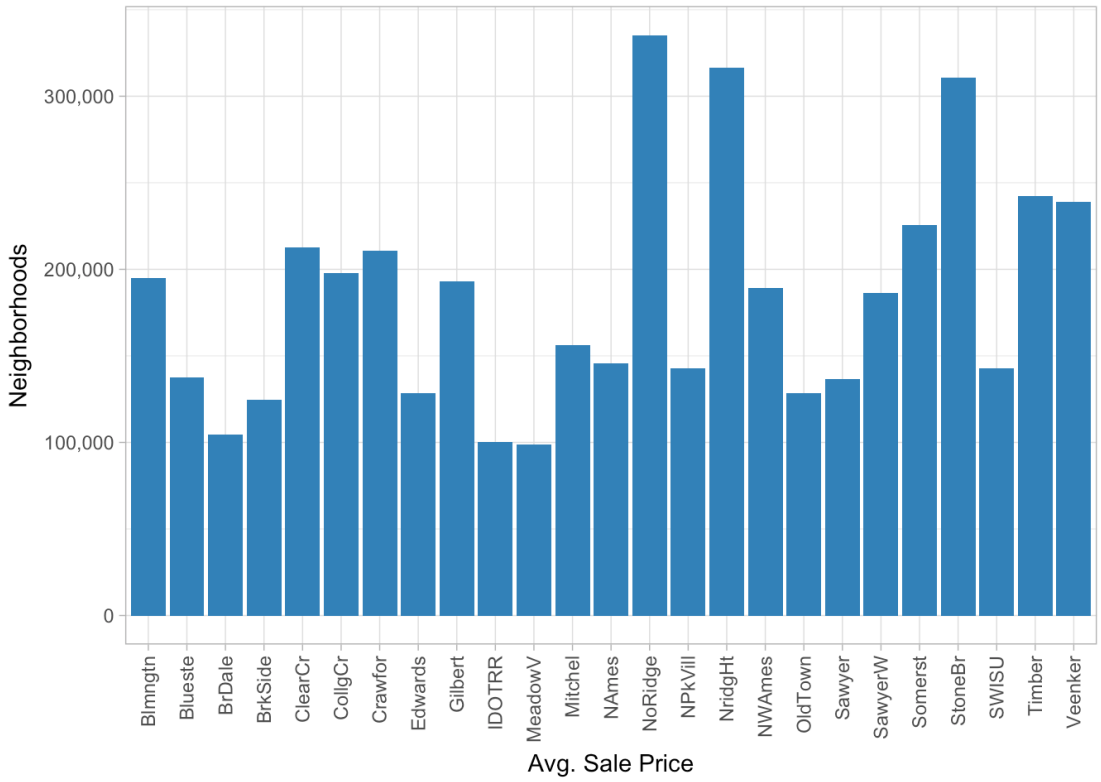

```
makeBarplots(train, train$LandContour, "Land Contour", "Count")
```




```
ggplot(train, aes(x=factor(Neighborhood))) + geom_bar(stat = "count", width=0.7, fill="steelblue") + ylab("N
eighborhoods") + xlab("Count") + theme_light() + theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0.5))
```

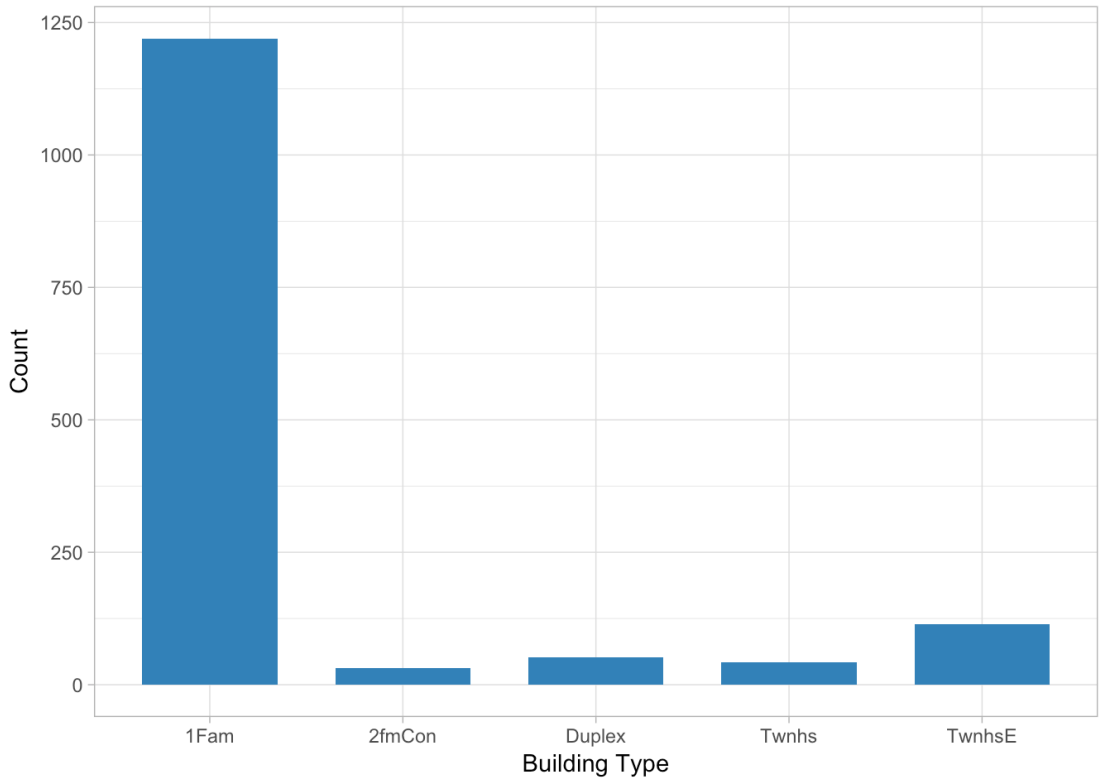



```
ggplot(train, aes(x=factor(Neighborhood),y=SalePrice)) + stat_summary(fun.y="mean", geom="bar", fill="steelb
lue") + ylab("Neighborhoods") + xlab("Avg. Sale Price") + theme_light() + theme(axis.text.x=element_text(ang
le=90,hjust=1,vjust=0.5)) + scale_y_continuous(labels = comma)
```

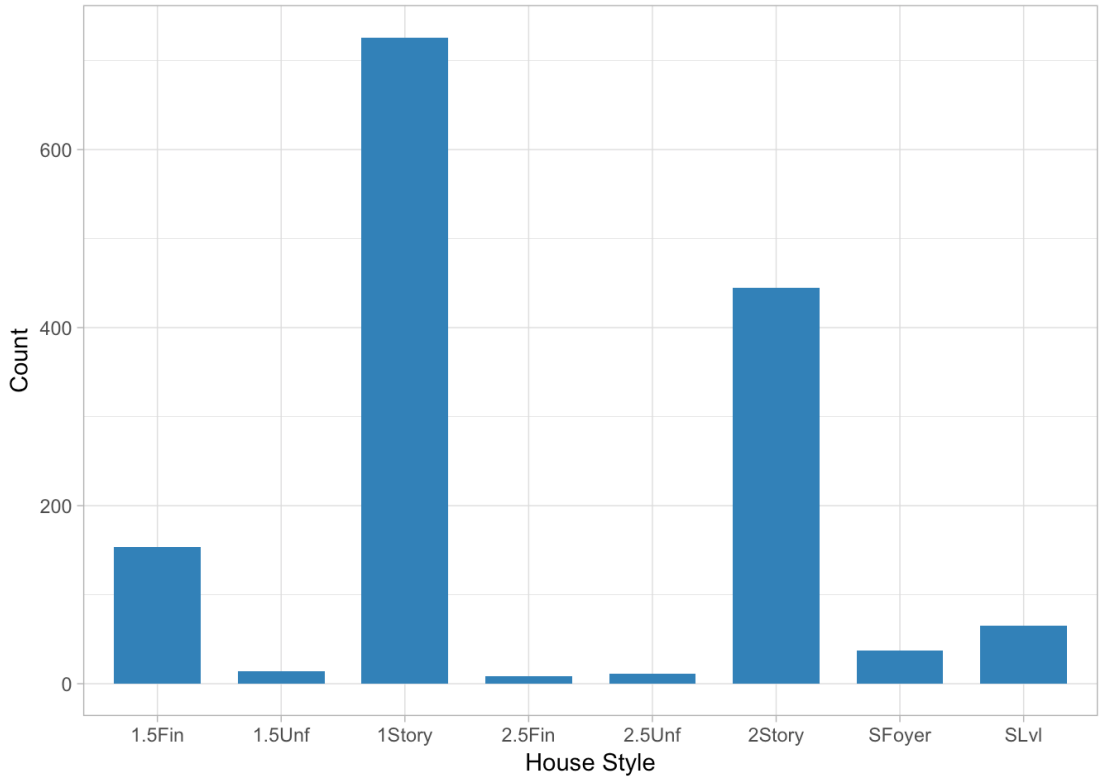


Area around the Iowa University, neighborhoods of College Creek and just N. Ames, just north of the university have high concentration of homes. The more affluent neighborhoods are NorthRidge, NorthRidge Heights and Stone Brook.

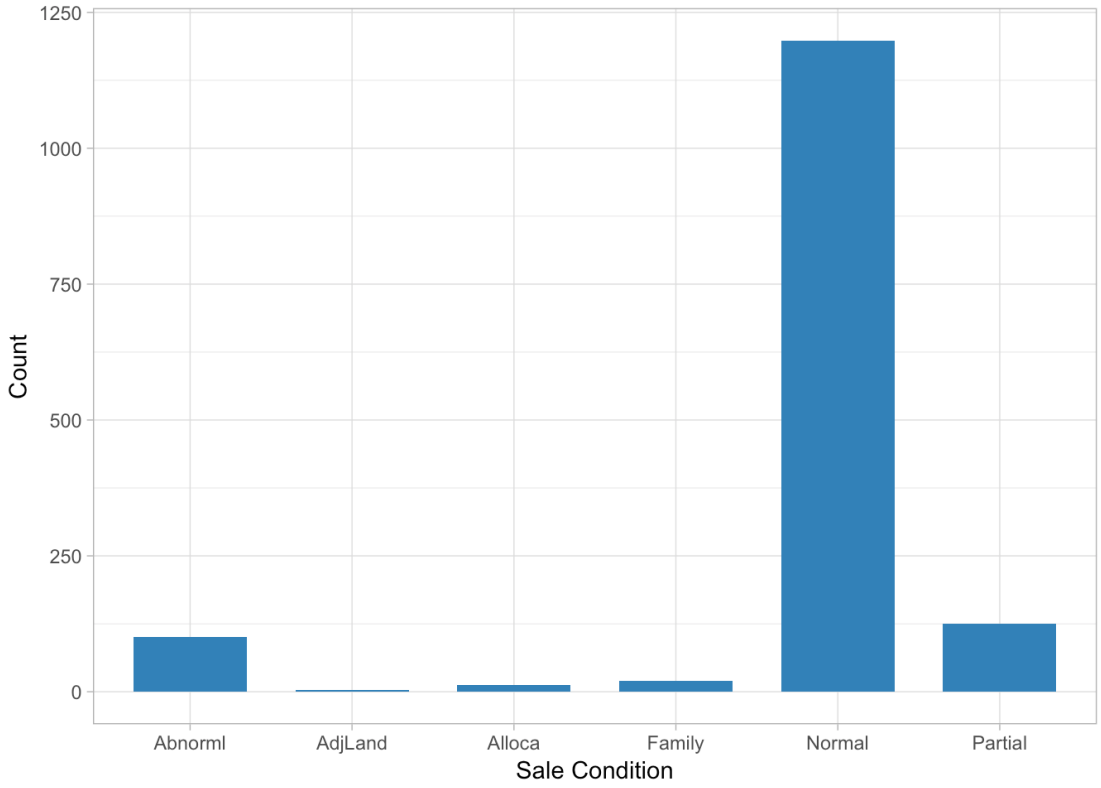
```
makeBarplots(train, train$BldgType, "Building Type", "Count")
```




```
makeBarplots(train, train$HouseStyle, "House Style", "Count")
```




```
makeBarplots(train, train$SaleCondition, "Sale Condition", "Count")
```



Let’s plot our dependent variable sales price

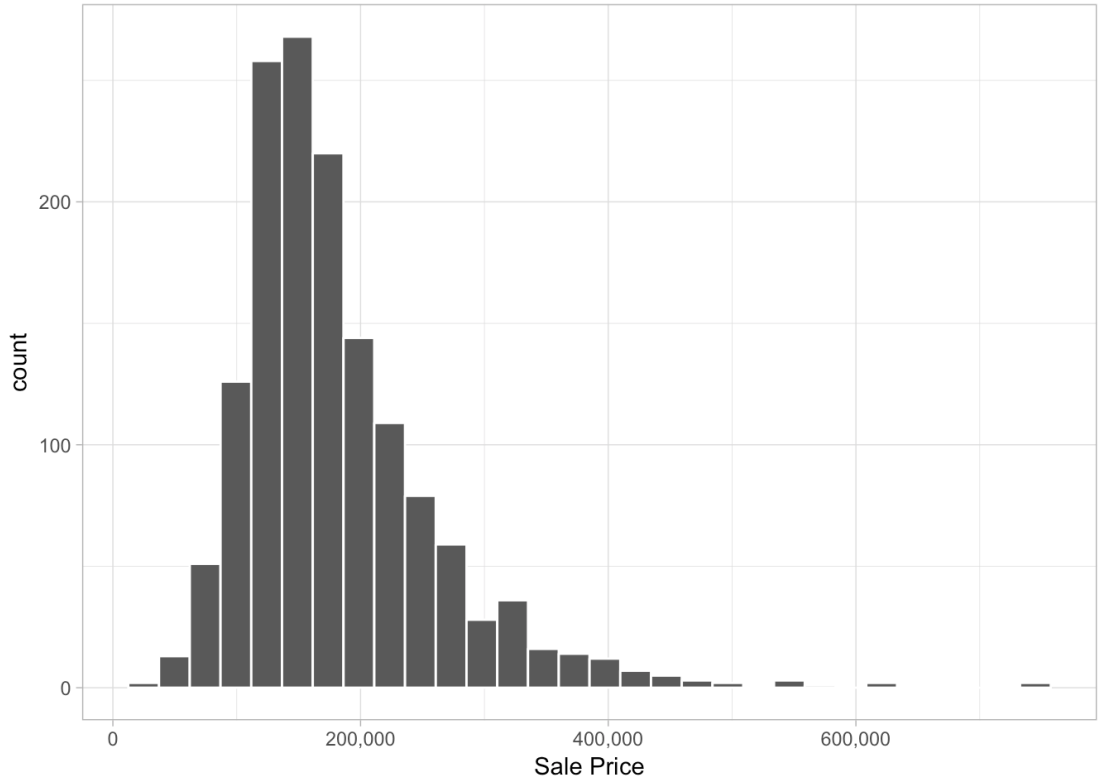
```
summary(train$SalePrice)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   34900  130000  163000  180900  214000  755000
```

```
ggplot(data=train, aes(train$SalePrice)) + geom_histogram(col = "white") + theme_light() + xlim(20000, 80000
0) + xlab("Sale Price") + scale_x_continuous(labels = comma)
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.
```

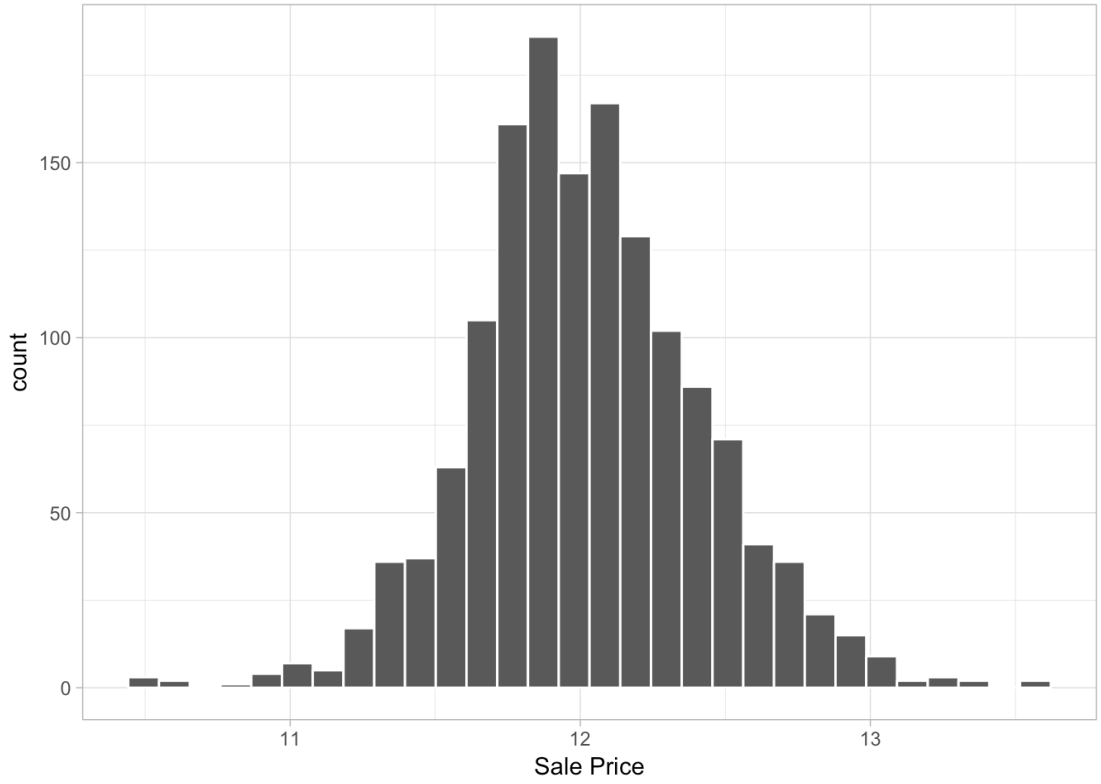
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Sale Price appears to be heavily skewed. We will log transform the variable to obtain a normal distribution of our dependent variable. This is to maintain positivity of the sale price variable, in all likelihood, sale price of a home will never be a negative value.

```
ggplot(data=train, aes(log(train$SalePrice))) + geom_histogram(col = "white") + theme_light() + xlab("Sale P
rice")
```

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




```
train$SalePrice = log(train$SalePrice + 1)
```

Before we get into building models, we will holdout the data from train set with sale price to be able to compare observed values with predictions.

```
response <- train$SalePrice

train_dummy <- dummy.data.frame(train, sep = ".", all = TRUE)
#names(train_dummy)

split <- createDataPartition(y=response,
                                  p=.5,
                                  list=F)
training <- train_dummy[split,]
testing <- train_dummy[-split,]

#str(training)
```

Let's build some advanced regression models.

Model building plan

First, we will build a simple linear regression to get a feel for the variables and relationship.

```
model.lm <- lm(SalePrice ~ ., data = training)
summary(model.lm)
```

```
##
## Call:
## lm(formula = SalePrice ~ ., data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.59982 -0.04274  0.00000  0.04597  0.59982
##
## Coefficients: (66 not defined because of singularities)
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.343e+00  7.753e+00   0.818 0.413662
## Id                     -1.568e-06  1.110e-05  -0.141 0.887693
## MSSubClass             -1.094e-03  8.028e-04  -1.363 0.173394
## `MSZoning.C (all)`     -4.447e-01  8.768e-02  -5.072 5.59e-07 ***
## MSZoning.FV             1.386e-01  5.576e-02   2.486 0.013240 *
## MSZoning.RH            -1.755e-02  5.526e-02  -0.318 0.750965
## MSZoning.RL             2.376e-02  2.933e-02   0.810 0.418307
## MSZoning.RM                    NA         NA      NA       NA
## LotFrontage             1.433e-04  3.044e-04   0.471 0.638001
## LotArea                 4.513e-06  1.603e-06   2.815 0.005081 **
## Street.Grvl            -6.197e-02  1.587e-01  -0.390 0.696354
## Street.Pave                    NA         NA      NA       NA
## Alley.Grvl             -2.365e-02  4.571e-02  -0.517 0.605119
## Alley.None              3.608e-02  3.519e-02   1.025 0.305779
## Alley.Pave                     NA         NA      NA       NA
## LotShape.IR1           -1.215e-02  1.174e-02  -1.035 0.301215
## LotShape.IR2            7.990e-03  3.216e-02   0.248 0.803917
## LotShape.IR3           -1.294e-02  6.389e-02  -0.203 0.839586
## LotShape.Reg                   NA         NA      NA       NA
## LandContour.Bnk        -7.802e-02  2.842e-02  -2.746 0.006262 **
## LandContour.HLS         5.547e-03  2.897e-02   0.191 0.848245
## LandContour.Low        -2.306e-02  4.875e-02  -0.473 0.636397
## LandContour.Lvl                NA         NA      NA       NA
## Utilities.AllPub        3.223e-01  1.494e-01   2.157 0.031513 *
## Utilities.NoSeWa               NA         NA      NA       NA
## LotConfig.Corner        3.214e-02  1.414e-02   2.272 0.023494 *
## LotConfig.CulDSac       5.007e-02  2.055e-02   2.437 0.015180 *
## LotConfig.FR2          -3.173e-02  2.748e-02  -1.154 0.248873
## LotConfig.FR3          -1.224e-01  8.205e-02  -1.491 0.136513
## LotConfig.Inside               NA         NA      NA       NA
## LandSlope.Gtl           8.281e-02  7.936e-02   1.043 0.297278
## LandSlope.Mod           1.135e-01  8.019e-02   1.415 0.157744
## LandSlope.Sev                  NA         NA      NA       NA
## Neighborhood.Blmngtn    4.528e-02  7.309e-02   0.619 0.535887
## Neighborhood.Blueste   -2.450e-02  1.066e-01  -0.230 0.818372
## Neighborhood.BrDale    -1.065e-01  7.636e-02  -1.394 0.163814
## Neighborhood.BrkSide   -7.479e-02  6.575e-02  -1.137 0.255907
## Neighborhood.ClearCr   -5.098e-02  5.717e-02  -0.892 0.372963
## Neighborhood.CollgCr   -5.505e-02  5.031e-02  -1.094 0.274364
## Neighborhood.Crawfor    1.141e-01  5.871e-02   1.944 0.052445 .
## Neighborhood.Edwards   -1.646e-01  5.259e-02  -3.131 0.001849 **
## Neighborhood.Gilbert   -3.761e-02  5.307e-02  -0.709 0.478803
## Neighborhood.IDOTRR    -1.198e-01  7.478e-02  -1.602 0.109857
## Neighborhood.MeadowV   -2.143e-01  7.927e-02  -2.704 0.007096 **
## Neighborhood.Mitchel   -4.919e-02  5.531e-02  -0.889 0.374268
## Neighborhood.NAmes     -1.004e-01  4.884e-02  -2.055 0.040381 *
## Neighborhood.NoRidge    1.524e-03  5.561e-02   0.027 0.978150
## Neighborhood.NPkVill   -1.052e-01  9.583e-02  -1.098 0.272723
## Neighborhood.NridgHt    8.545e-02  5.559e-02   1.537 0.124873
## Neighborhood.NWAmes    -8.604e-02  5.135e-02  -1.676 0.094437 .
## Neighborhood.OldTown   -1.217e-01  6.146e-02  -1.980 0.048299 *
## Neighborhood.Sawyer    -9.079e-02  5.039e-02  -1.802 0.072189 .
## Neighborhood.SawyerW   -3.654e-02  5.092e-02  -0.718 0.473308
## Neighborhood.Somerst   -8.050e-02  6.490e-02  -1.240 0.215447
## Neighborhood.StoneBr    1.067e-01  5.947e-02   1.794 0.073429 .
```

```
## Neighborhood.SWISU    -1.008e-01  6.745e-02  -1.494 0.135730
## Neighborhood.Timber   -5.133e-02  5.673e-02  -0.905 0.365966
## Neighborhood.Veenker          NA         NA      NA       NA
## Condition1.Artery     -1.459e-01  1.025e-01  -1.424 0.155134
## Condition1.Feedr      -1.058e-01  9.779e-02  -1.082 0.279611
## Condition1.Norm       -6.741e-02  9.597e-02  -0.702 0.482751
## Condition1.PosA       -1.245e-01  1.110e-01  -1.121 0.262710
## Condition1.PosN       -1.152e-02  1.069e-01  -0.108 0.914227
## Condition1.RRAe       -2.769e-01  1.152e-01  -2.403 0.016615 *
## Condition1.RRAn       -9.572e-02  1.027e-01  -0.932 0.351619
## Condition1.RRNe       -1.014e-01  1.485e-01  -0.683 0.494996
## Condition1.RRNn               NA         NA      NA       NA
## Condition2.Artery     -4.237e-02  1.830e-01  -0.231 0.817052
## Condition2.Feedr       1.052e-01  1.580e-01   0.666 0.505851
## Condition2.Norm       -2.602e-02  1.276e-01  -0.204 0.838515
## Condition2.PosA        1.269e-01  2.061e-01   0.615 0.538511
## Condition2.PosN       -8.822e-01  1.652e-01  -5.340 1.42e-07 ***
## Condition2.RRAe       -1.935e-01  4.847e-01  -0.399 0.689827
## Condition2.RRAn               NA         NA      NA       NA
## Condition2.RRNn               NA         NA      NA       NA
## BldgType.1Fam         -6.457e-02  8.402e-02  -0.769 0.442560
## BldgType.2fmCon        6.012e-02  6.875e-02   0.874 0.382286
## BldgType.Duplex       -1.119e-01  8.183e-02  -1.368 0.171956
## BldgType.Twnhs        -6.031e-02  3.677e-02  -1.640 0.101570
## BldgType.TwnhsE               NA         NA      NA       NA
## HouseStyle.1.5Fin     -7.847e-03  4.237e-02  -0.185 0.853164
## HouseStyle.1.5Unf     -6.005e-02  1.122e-01  -0.535 0.592808
## HouseStyle.1Story     -5.548e-02  5.302e-02  -1.046 0.295893
## HouseStyle.2.5Fin     -1.012e-02  9.725e-02  -0.104 0.917192
## HouseStyle.2.5Unf      4.809e-02  6.946e-02   0.692 0.489012
## HouseStyle.2Story     -5.916e-02  3.887e-02  -1.522 0.128675
## HouseStyle.SFoyer      4.046e-04  3.734e-02   0.011 0.991360
## HouseStyle.SLvl               NA         NA      NA       NA
## OverallQual            4.231e-02  7.686e-03   5.505 5.98e-08 ***
## OverallCond            3.910e-02  6.704e-03   5.832 9.94e-09 ***
## YearBuilt              1.537e-03  6.140e-04   2.502 0.012659 *
## YearRemodAdd           9.808e-04  4.600e-04   2.132 0.033485 *
## RoofStyle.Flat        -3.549e-01  2.101e-01  -1.689 0.091785 .
## RoofStyle.Gable       -3.246e-01  1.707e-01  -1.902 0.057731 .
## RoofStyle.Gambrel     -3.884e-01  1.853e-01  -2.096 0.036558 *
## RoofStyle.Hip         -3.206e-01  1.707e-01  -1.878 0.061039 .
## RoofStyle.Mansard     -3.666e-01  1.664e-01  -2.204 0.028012 *
## RoofStyle.Shed                NA         NA      NA       NA
## RoofMatl.ClyTile              NA         NA      NA       NA
## RoofMatl.CompShg      -8.031e-02  1.236e-01  -0.650 0.516284
## RoofMatl.Membran       1.853e-01  2.360e-01   0.785 0.432677
## RoofMatl.Metal         5.042e-02  2.277e-01   0.221 0.824824
## RoofMatl.Roll                 NA         NA      NA       NA
## `RoofMatl.Tar&Grv`    -6.039e-02  1.747e-01  -0.346 0.729667
## RoofMatl.WdShake      -8.445e-02  1.508e-01  -0.560 0.575624
## RoofMatl.WdShngl              NA         NA      NA       NA
## Exterior1st.AsbShng    5.156e-02  8.115e-02   0.635 0.525520
## Exterior1st.AsphShn   -2.342e-01  2.137e-01  -1.096 0.273670
## Exterior1st.BrkComm   -4.390e-01  1.909e-01  -2.300 0.021859 *
## Exterior1st.BrkFace    3.967e-02  6.370e-02   0.623 0.533695
## Exterior1st.CBlock            NA         NA      NA       NA
## Exterior1st.CemntBd   -2.661e-02  1.363e-01  -0.195 0.845281
## Exterior1st.HdBoard   -7.248e-02  6.036e-02  -1.201 0.230449
## Exterior1st.ImStucc           NA         NA      NA       NA
## Exterior1st.MetalSd   -3.432e-03  7.015e-02  -0.049 0.961001
## Exterior1st.Plywood   -6.154e-02  5.861e-02  -1.050 0.294232
## Exterior1st.Stone     -2.134e-02  1.684e-01  -0.127 0.899248
## Exterior1st.Stucco     4.812e-02  1.075e-01   0.448 0.654536
## Exterior1st.VinylSd   -9.261e-02  7.587e-02  -1.221 0.222833
## `Exterior1st.Wd Sdng` -5.747e-02  5.948e-02  -0.966 0.334362
## Exterior1st.WdShing           NA         NA      NA       NA
```

```
## Exterior2nd.AsbShng           NA         NA      NA       NA
## Exterior2nd.AsphShn    2.045e-01  1.580e-01   1.294 0.196159
## `Exterior2nd.Brk Cmn`  8.959e-02  1.228e-01   0.730 0.465856
## Exterior2nd.BrkFace   -4.328e-03  6.042e-02  -0.072 0.942929
## Exterior2nd.CBlock            NA         NA      NA       NA
## Exterior2nd.CmentBd    2.456e-02  1.356e-01   0.181 0.856325
## Exterior2nd.HdBoard    3.173e-02  5.072e-02   0.626 0.531846
## Exterior2nd.ImStucc    6.279e-02  7.509e-02   0.836 0.403502
## Exterior2nd.MetalSd    2.968e-03  6.018e-02   0.049 0.960686
## Exterior2nd.Other             NA         NA      NA       NA
## Exterior2nd.Plywood    3.250e-02  4.909e-02   0.662 0.508226
## Exterior2nd.Stone      1.181e-01  1.679e-01   0.703 0.482342
## Exterior2nd.Stucco     5.888e-02  9.260e-02   0.636 0.525158
## Exterior2nd.VinylSd    7.448e-02  6.080e-02   1.225 0.221104
## `Exterior2nd.Wd Sdng`  5.535e-02  4.305e-02   1.286 0.199108
## `Exterior2nd.Wd Shng`         NA         NA      NA       NA
## MasVnrType.BrkCmn     -9.037e-02  5.714e-02  -1.582 0.114405
## MasVnrType.BrkFace     2.066e-03  2.272e-02   0.091 0.927610
## MasVnrType.None       -2.387e-02  2.359e-02  -1.012 0.312162
## MasVnrType.Stone              NA         NA      NA       NA
## MasVnrArea            -3.469e-06  4.052e-05  -0.086 0.931800
## ExterQual.Ex          -2.192e-02  4.038e-02  -0.543 0.587529
## ExterQual.Fa           9.218e-02  9.650e-02   0.955 0.339900
## ExterQual.Gd          -2.769e-02  1.794e-02  -1.543 0.123444
## ExterQual.TA                  NA         NA      NA       NA
## ExterCond.Ex                  NA         NA      NA       NA
## ExterCond.Fa          -1.431e-02  4.946e-02  -0.289 0.772444
## ExterCond.Gd          -5.205e-02  1.829e-02  -2.846 0.004606 **
## ExterCond.Po           2.145e-03  1.541e-01   0.014 0.988899
## ExterCond.TA                  NA         NA      NA       NA
## Foundation.BrkTil      1.715e-01  1.383e-01   1.240 0.215548
## Foundation.CBlock      2.392e-01  1.360e-01   1.759 0.079192 .
## Foundation.PConc       2.513e-01  1.352e-01   1.859 0.063697 .
## Foundation.Slab        1.528e-01  1.533e-01   0.996 0.319544
## Foundation.Stone       1.819e-01  2.143e-01   0.849 0.396309
## Foundation.Wood               NA         NA      NA       NA
## BsmtQual.Ex            4.152e-02  2.937e-02   1.414 0.158085
## BsmtQual.Fa           -2.466e-02  4.441e-02  -0.555 0.579026
## BsmtQual.Gd            5.819e-03  1.838e-02   0.317 0.751647
## BsmtQual.None          2.408e-01  1.930e-01   1.248 0.212807
## BsmtQual.TA                   NA         NA      NA       NA
## BsmtCond.Fa           -2.075e-02  3.261e-02  -0.636 0.524751
## BsmtCond.Gd            1.929e-02  2.414e-02   0.799 0.424671
## BsmtCond.None                 NA         NA      NA       NA
## BsmtCond.Po            9.225e-02  1.895e-01   0.487 0.626632
## BsmtCond.TA                   NA         NA      NA       NA
## BsmtExposure.Av        2.592e-02  1.105e-01   0.235 0.814661
## BsmtExposure.Gd        8.262e-02  1.117e-01   0.740 0.459841
## BsmtExposure.Mn        5.331e-02  1.115e-01   0.478 0.632802
## BsmtExposure.No        3.524e-02  1.101e-01   0.320 0.749109
## BsmtExposure.None             NA         NA      NA       NA
## BsmtFinType1.ALQ       2.404e-04  2.198e-02   0.011 0.991278
## BsmtFinType1.BLQ       4.419e-03  2.398e-02   0.184 0.853884
## BsmtFinType1.GLQ       3.137e-02  2.025e-02   1.549 0.122024
## BsmtFinType1.LwQ      -8.747e-03  2.888e-02  -0.303 0.762116
## BsmtFinType1.None             NA         NA      NA       NA
## BsmtFinType1.Rec       6.354e-03  2.425e-02   0.262 0.793414
## BsmtFinType1.Unf              NA         NA      NA       NA
## BsmtFinSF1             1.448e-04  4.128e-05   3.509 0.000492 ***
## BsmtFinType2.ALQ      -5.693e-03  6.418e-02  -0.089 0.929363
## BsmtFinType2.BLQ      -3.688e-02  3.915e-02  -0.942 0.346569
## BsmtFinType2.GLQ       1.037e-03  6.375e-02   0.016 0.987033
## BsmtFinType2.LwQ      -2.727e-02  3.184e-02  -0.857 0.392021
## BsmtFinType2.None     -1.605e-01  1.303e-01  -1.232 0.218428
## BsmtFinType2.Rec      -7.168e-03  3.441e-02  -0.208 0.835094
## BsmtFinType2.Unf              NA         NA      NA       NA
```

```
## BsmtFinSF2             1.548e-04  6.984e-05   2.217 0.027109 *
## BsmtUnfSF              9.026e-05  3.734e-05   2.417 0.016002 *
## TotalBsmtSF                   NA         NA      NA       NA
## Heating.Floor         -1.729e-01  1.808e-01  -0.956 0.339480
## Heating.GasA          -1.042e-02  1.104e-01  -0.094 0.924827
## Heating.GasW           1.104e-01  1.277e-01   0.864 0.387742
## Heating.Grav          -1.092e-01  1.417e-01  -0.771 0.441194
## Heating.OthW           4.078e-02  1.518e-01   0.269 0.788345
## Heating.Wall                  NA         NA      NA       NA
## HeatingQC.Ex           5.485e-02  1.633e-02   3.358 0.000847 ***
## HeatingQC.Fa           1.934e-02  3.326e-02   0.581 0.561310
## HeatingQC.Gd           2.214e-02  1.671e-02   1.325 0.185834
## HeatingQC.Po                  NA         NA      NA       NA
## HeatingQC.TA                  NA         NA      NA       NA
## CentralAir.N          -2.549e-02  3.065e-02  -0.832 0.406049
## CentralAir.Y                  NA         NA      NA       NA
## Electrical.FuseA      -1.515e-02  2.200e-02  -0.689 0.491291
## Electrical.FuseF      -1.471e-01  5.211e-02  -2.823 0.004948 **
## Electrical.FuseP       3.264e-02  2.911e-01   0.112 0.910751
## Electrical.Mix         1.987e-01  3.041e-01   0.653 0.513805
## Electrical.None        9.037e-02  1.205e-01   0.750 0.453524
## Electrical.SBrkr              NA         NA      NA       NA
## X1stFlrSF              1.460e-04  4.238e-05   3.445 0.000620 ***
## X2ndFlrSF              1.879e-04  4.094e-05   4.588 5.68e-06 ***
## LowQualFinSF           1.590e-04  1.224e-04   1.299 0.194478
## GrLivArea                     NA         NA      NA       NA
## BsmtFullBath           1.841e-02  1.497e-02   1.230 0.219451
## BsmtHalfBath           1.845e-03  2.220e-02   0.083 0.933792
## FullBath               3.571e-02  1.679e-02   2.127 0.033928 *
## HalfBath               2.654e-02  1.527e-02   1.738 0.082921 .
## BedroomAbvGr          -6.451e-04  9.989e-03  -0.065 0.948534
## KitchenAbvGr           1.007e-02  6.286e-02   0.160 0.872786
## KitchenQual.Ex         8.390e-02  2.866e-02   2.927 0.003577 **
## KitchenQual.Fa        -1.495e-03  3.905e-02  -0.038 0.969479
## KitchenQual.Gd         1.236e-02  1.580e-02   0.782 0.434475
## KitchenQual.TA                NA         NA      NA       NA
## TotRmsAbvGrd           1.824e-02  7.273e-03   2.508 0.012476 *
## Functional.Maj1       -4.752e-02  4.424e-02  -1.074 0.283361
## Functional.Maj2       -3.951e-01  8.329e-02  -4.743 2.77e-06 ***
## Functional.Min1       -4.764e-02  3.282e-02  -1.452 0.147255
## Functional.Min2       -2.139e-02  3.367e-02  -0.635 0.525474
## Functional.Mod        -1.315e-01  6.732e-02  -1.953 0.051414 .
## Functional.Sev        -3.149e-01  1.653e-01  -1.905 0.057388 .
## Functional.Typ                NA         NA      NA       NA
## Fireplaces             2.892e-02  1.807e-02   1.600 0.110246
## FireplaceQu.Ex        -8.462e-03  4.347e-02  -0.195 0.845732
## FireplaceQu.Fa         1.260e-04  3.029e-02   0.004 0.996682
## FireplaceQu.Gd        -2.180e-02  1.655e-02  -1.317 0.188572
## FireplaceQu.None       4.309e-03  2.547e-02   0.169 0.865734
## FireplaceQu.Po         6.273e-02  4.381e-02   1.432 0.152836
## FireplaceQu.TA                NA         NA      NA       NA
## GarageType.2Types      1.552e+00  9.474e-01   1.638 0.102062
## GarageType.Attchd      1.624e+00  9.392e-01   1.729 0.084488 .
## GarageType.Basment     1.666e+00  9.358e-01   1.781 0.075606 .
## GarageType.BuiltIn     1.609e+00  9.412e-01   1.710 0.087967 .
## GarageType.CarPort     1.690e+00  9.471e-01   1.785 0.074925 .
## GarageType.Detchd      1.640e+00  9.420e-01   1.741 0.082250 .
## GarageType.None               NA         NA      NA       NA
## GarageYrBlt           -8.378e-04  4.820e-04  -1.738 0.082831 .
## GarageFinish.Fin       1.990e-02  1.779e-02   1.118 0.263961
## GarageFinish.None             NA         NA      NA       NA
## GarageFinish.RFn       1.739e-02  1.579e-02   1.101 0.271478
## GarageFinish.Unf              NA         NA      NA       NA
## GarageCars             4.642e-03  1.780e-02   0.261 0.794385
## GarageArea             1.937e-04  6.177e-05   3.136 0.001815 **
## GarageQual.Ex                 NA         NA      NA       NA
```

```
## GarageQual.Fa         -4.525e-02  3.438e-02  -1.316 0.188708
## GarageQual.Gd          3.605e-02  6.741e-02   0.535 0.593064
## GarageQual.None               NA         NA      NA       NA
## GarageQual.Po         -2.104e-01  2.224e-01  -0.946 0.344614
## GarageQual.TA                 NA         NA      NA       NA
## GarageCond.Ex                 NA         NA      NA       NA
## GarageCond.Fa         -6.437e-02  4.194e-02  -1.535 0.125483
## GarageCond.Gd         -1.422e-02  7.522e-02  -0.189 0.850197
## GarageCond.None               NA         NA      NA       NA
## GarageCond.Po          1.150e-01  1.292e-01   0.890 0.373746
## GarageCond.TA                 NA         NA      NA       NA
## PavedDrive.N          -3.302e-02  2.976e-02  -1.109 0.267795
## PavedDrive.P          -2.195e-02  3.818e-02  -0.575 0.565685
## PavedDrive.Y                  NA         NA      NA       NA
## WoodDeckSF             1.003e-04  4.252e-05   2.359 0.018726 *
## OpenPorchSF            1.337e-04  8.610e-05   1.553 0.121086
## EnclosedPorch          2.311e-04  9.331e-05   2.476 0.013609 *
## X3SsnPorch             3.104e-04  1.836e-04   1.691 0.091502 .
## ScreenPorch            2.458e-04  8.622e-05   2.851 0.004537 **
## PoolArea               2.058e-04  2.180e-04   0.944 0.345768
## PoolQC.Ex                     NA         NA      NA       NA
## PoolQC.Fa             -2.666e-03  1.860e-01  -0.014 0.988571
## PoolQC.Gd                     NA         NA      NA       NA
## PoolQC.None                   NA         NA      NA       NA
## Fence.GdPrv           -4.643e-02  2.578e-02  -1.801 0.072352 .
## Fence.GdWo            -1.996e-02  2.639e-02  -0.756 0.449961
## Fence.MnPrv            1.594e-03  1.665e-02   0.096 0.923766
## Fence.MnWw            -5.021e-02  5.683e-02  -0.883 0.377425
## Fence.None                    NA         NA      NA       NA
## MiscFeature.Gar2      -7.397e-01  8.348e-01  -0.886 0.376011
## MiscFeature.None       5.450e-02  4.713e-02   1.156 0.248094
## MiscFeature.Othr       1.502e-01  1.287e-01   1.167 0.243936
## MiscFeature.Shed              NA         NA      NA       NA
## MiscFeature.TenC              NA         NA      NA       NA
## MiscVal                5.440e-05  5.583e-05   0.975 0.330282
## MoSold                 8.198e-04  1.801e-03   0.455 0.649167
## YrSold                -2.482e-04  3.822e-03  -0.065 0.948252
## SaleType.COD           3.798e-02  2.886e-02   1.316 0.188827
## SaleType.Con           1.786e-03  1.140e-01   0.016 0.987508
## SaleType.ConLD         1.128e-01  6.704e-02   1.682 0.093203 .
## SaleType.ConLI        -5.750e-03  7.103e-02  -0.081 0.935520
## SaleType.ConLw         2.555e-02  9.359e-02   0.273 0.784945
## SaleType.CWD           4.337e-02  7.604e-02   0.570 0.568642
## SaleType.New          -1.636e-02  1.375e-01  -0.119 0.905342
## SaleType.Oth           2.708e-01  1.268e-01   2.137 0.033133 *
## SaleType.WD                   NA         NA      NA       NA
## SaleCondition.Abnorml -1.265e-01  1.380e-01  -0.917 0.359686
## SaleCondition.AdjLand -9.003e-02  1.950e-01  -0.462 0.644479
## SaleCondition.Alloca  -1.218e-01  1.632e-01  -0.746 0.455922
## SaleCondition.Family  -1.931e-01  1.436e-01  -1.344 0.179462
## SaleCondition.Normal  -6.043e-02  1.376e-01  -0.439 0.660618
## SaleCondition.Partial         NA         NA      NA       NA
## Remodel_flag.No        1.076e-02  1.270e-02   0.847 0.397280
## Remodel_flag.Yes              NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1061 on 490 degrees of freedom
## Multiple R-squared:  0.9522, Adjusted R-squared:  0.9287
## F-statistic: 40.64 on 240 and 490 DF,  p-value: < 2.2e-16
```

Our R-Squared of 0.93 is not bad at all. Looking at the coefficients and their corresponding values, we see

there

are lots of predictors that we can drop or are not significant. The F-Statistic of 45 shows that there is relationship between the response variable - ‘SalePrice’ and predictors.

Quick side note: Referencing and cross checking, highly correlated variables with SalePrice in our correlation plot above and simple linear regression, we can be assured that the highly correlated variables are indeed significant variables.

We noticed in our linear regression output that some variables are set to NA. These are set to NA as they don’t add any extra value because of multi-collinearity.

Our model spits out 307 predictor variables, that’s a lot of variables to sort through and drop non-significant one’s.

Let’s perform PCA to reduce some of the features from our model.

```
# Principal component analysis

# PCA works well on normalized dataset.
# This is because there could be large loadings due to the way variables are measured.
#training.scaled <- data.frame(apply(training, 2, scale))

# Remove missing values or NAs
# sum(is.na(training.scaled))
#training.scale.na.omit <- data.frame(t(na.omit(t(training.scaled))))

# Run PCA
#training_pca <- prcomp(training.scale.na.omit, retx=TRUE)
#names(training_pca)
#training_pca$center
#training_pca$scale
#training_pca$rotation
#dim(training_pca$x)
```

This returns 286 principal component loadings.

The maximum number of principal component loadings is a minimum of (n-1, p).

```
# Plot
#biplot(training_pca, choices=1:2, scale = 0)
```

```
#summary(training_pca)
```

The 1 PC explains 6.8%, 2 PC explains 3.1% of variance in the data and so on.

```
# Calculate Variance
#pr_var <- training_pca$sdev^2

#plot(pr_var, type = "l", xlab = "Principapl Components", ylab = "Proportion of Variance explained")

#training_pca$rotation
```

The plot method returns a plot of the variances (y-axis) associated with the PCs (x-axis). The Figure below is useful to decide how many PCs to retain for further analysis.

The plot above shows that ~40 Principal components explain most of the variance (80% +) in the data. PCA has helped us reduce 307 explanatory variables to 40 without compromising on variance.

Now that we've computed the Principal components on training data, we will use these components to predict on test data.

```
# Transformation similar to training set.

#Add a training set with principal components
#training.data.pca <- data.frame(training$SalePrice, training_pca$x)

# Extract first 40 Principal Components
#training.data.pca <- training.data.pca[,1:40]
```

Run a linear regression with PCA transformed data

As I was researching some of the regression techniques that I can apply on PCA transformed data, turns out there are some other techniques that we can possibly utilize. See more details in the response to the question: http://stats.stackexchange.com/questions/269032/pca-transformed-data-and-regression (http://stats.stackexchange.com/questions/269032/pca-transformed-data-and-regression).

We will use the Partial Least Squares Regression which combines PCA and Regression with loadings that are also highly correlated with the response variable. Since, we are interested in predicting SalePrice, PCR seems like a more logical choice.

```
#require(pls)

#pcr.model <- pcr(SalePrice ~ ., data = train, scale = TRUE, validation = "CV")
#summary(pcr.model)
```

```
response1 <- train$SalePrice

split <- createDataPartition(y=response1,
                                p=.5,
                                list=F)
training1 <- train[split,]
testing1 <- train[-split,]
```

Feature Engineering, based on some documentation on ridge vs lasso here and elsewhere, http://stats.stackexchange.com/questions/866/when-should-i-use-lasso-vs-ridge (http://stats.stackexchange.com/questions/866/when-should-i-use-lasso-vs-ridge), we will try lasso for feature selection. Given that we have a large number of parameter estimates and as our results from simple linear regression indicate that not all variables are correlated with the response variable SalePrice. Lasso

seems to be an appropriate choice here for feature engineering/selection.

```
library(glmnet)
```

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

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-5
```

```
train1 = training1[,2:81]
#str(train1)

# glmnet requires a matrix of predictors and a response variable

x = model.matrix(SalePrice ~., data = train1)
#dim(x)

y = train1$SalePrice
```

Cross Validation to find optimal value of Lambda. CV is a predictive criterion that evaluates the sample performance by splitting the sample into training and validation sets and choosing the value of lambda with which the error of prediction is minimal.

```
cv.lasso <- cv.glmnet(x, y)
print(cv.lasso)
```

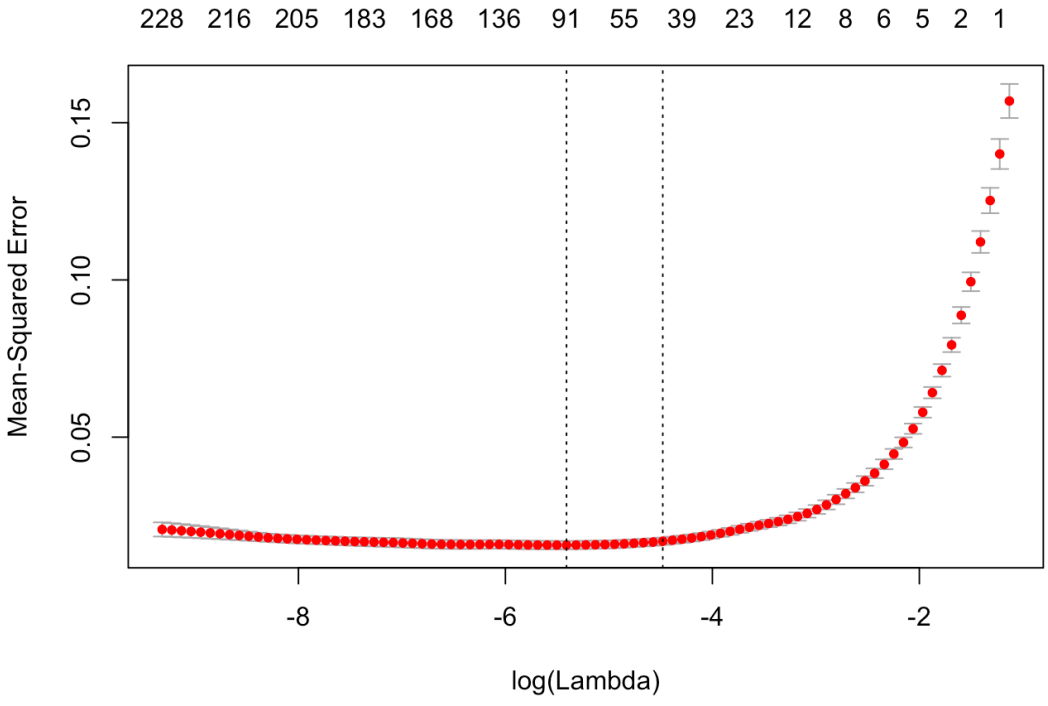
```
## $lambda
##  [1] 3.230609e-01 2.943610e-01 2.682108e-01 2.443837e-01 2.226733e-01
##  [6] 2.028916e-01 1.848673e-01 1.684442e-01 1.534801e-01 1.398453e-01
## [11] 1.274219e-01 1.161021e-01 1.057879e-01 9.638996e-02 8.782695e-02
## [16] 8.002464e-02 7.291547e-02 6.643786e-02 6.053571e-02 5.515788e-02
## [21] 5.025781e-02 4.579304e-02 4.172491e-02 3.801819e-02 3.464076e-02
## [26] 3.156337e-02 2.875937e-02 2.620446e-02 2.387653e-02 2.175541e-02
## [31] 1.982271e-02 1.806172e-02 1.645717e-02 1.499516e-02 1.366303e-02
## [36] 1.244924e-02 1.134329e-02 1.033558e-02 9.417396e-03 8.580780e-03
## [41] 7.818487e-03 7.123914e-03 6.491045e-03 5.914399e-03 5.388980e-03
## [46] 4.910238e-03 4.474026e-03 4.076566e-03 3.714415e-03 3.384436e-03
## [51] 3.083772e-03 2.809819e-03 2.560202e-03 2.332761e-03 2.125525e-03
## [56] 1.936699e-03 1.764648e-03 1.607882e-03 1.465042e-03 1.334891e-03
## [61] 1.216303e-03 1.108250e-03 1.009796e-03 9.200889e-04 8.383508e-04
## [66] 7.638740e-04 6.960135e-04 6.341816e-04 5.778427e-04 5.265087e-04
## [71] 4.797351e-04 4.371168e-04 3.982845e-04 3.629020e-04 3.306628e-04
## [76] 3.012876e-04 2.745221e-04 2.501343e-04 2.279131e-04 2.076659e-04
## [81] 1.892174e-04 1.724079e-04 1.570916e-04 1.431360e-04 1.304202e-04
## [86] 1.188341e-04 1.082772e-04 9.865812e-05 8.989360e-05
##
## $cvm
##  [1] 0.15690584 0.14005429 0.12525225 0.11207856 0.09942188 0.08877025
##  [7] 0.07933219 0.07123502 0.06414553 0.05790818 0.05267251 0.04832421
## [13] 0.04464719 0.04135563 0.03850567 0.03607323 0.03392086 0.03202438
## [19] 0.03018021 0.02846723 0.02699243 0.02575835 0.02474532 0.02385980
## [25] 0.02313329 0.02253748 0.02195603 0.02134226 0.02068991 0.02003125
## [31] 0.01942019 0.01886031 0.01836128 0.01792508 0.01754240 0.01720495
## [37] 0.01689551 0.01663713 0.01643774 0.01625512 0.01608753 0.01593939
## [43] 0.01584060 0.01578141 0.01572795 0.01568251 0.01566022 0.01566591
## [49] 0.01568033 0.01569113 0.01571851 0.01577342 0.01583664 0.01588765
## [55] 0.01587805 0.01585963 0.01585476 0.01586127 0.01587502 0.01592244
## [61] 0.01599496 0.01610453 0.01622393 0.01634867 0.01647143 0.01655119
## [67] 0.01663296 0.01671947 0.01680310 0.01689772 0.01701795 0.01713262
## [73] 0.01724304 0.01736892 0.01749717 0.01764575 0.01781956 0.01800373
## [79] 0.01824186 0.01849614 0.01875371 0.01901870 0.01927921 0.01953177
## [85] 0.01978062 0.02001452 0.02023528 0.02045840 0.02063917
##
## $cvsd
##  [1] 0.005414122 0.004773816 0.004038831 0.003479880 0.002993954
##  [6] 0.002622653 0.002276630 0.001991604 0.001795843 0.001690317
## [11] 0.001633542 0.001604116 0.001595247 0.001563680 0.001538869
## [16] 0.001522489 0.001501212 0.001504799 0.001515717 0.001467940
## [21] 0.001412176 0.001364131 0.001318191 0.001266810 0.001221935
## [26] 0.001183068 0.001152983 0.001133926 0.001111654 0.001108090
## [31] 0.001107183 0.001106902 0.001116315 0.001121281 0.001126766
## [36] 0.001134863 0.001149010 0.001166167 0.001192948 0.001221153
## [41] 0.001242859 0.001254437 0.001267420 0.001275914 0.001284092
## [46] 0.001290794 0.001297890 0.001308278 0.001317644 0.001321914
## [51] 0.001329415 0.001341245 0.001349937 0.001361512 0.001361035
## [56] 0.001366795 0.001381594 0.001389597 0.001394748 0.001406597
## [61] 0.001410910 0.001420923 0.001430829 0.001439887 0.001447981
## [66] 0.001421638 0.001385078 0.001349215 0.001325057 0.001312069
## [71] 0.001304391 0.001295105 0.001285343 0.001278481 0.001273309
## [76] 0.001269488 0.001273429 0.001290468 0.001351531 0.001435086
## [81] 0.001528628 0.001630068 0.001728269 0.001824658 0.001923199
## [86] 0.002016413 0.002102793 0.002178822 0.002252824
##
## $cvup
##  [1] 0.16231996 0.14482811 0.12929108 0.11555844 0.10241583 0.09139290
##  [7] 0.08160882 0.07322663 0.06594137 0.05959850 0.05430605 0.04992832
## [13] 0.04624244 0.04291931 0.04004454 0.03759572 0.03542207 0.03352918
## [19] 0.03169593 0.02993517 0.02840461 0.02712248 0.02606351 0.02512661
## [25] 0.02435523 0.02372055 0.02310901 0.02247619 0.02180156 0.02113934
## [31] 0.02052737 0.01996721 0.01947760 0.01904636 0.01866916 0.01833981
## [37] 0.01804452 0.01780330 0.01763069 0.01747627 0.01733039 0.01719383
```

```
## [43] 0.01710802 0.01705732 0.01701204 0.01697330 0.01695811 0.01697419
## [49] 0.01699797 0.01701305 0.01704792 0.01711466 0.01718658 0.01724916
## [55] 0.01723909 0.01722643 0.01723635 0.01725087 0.01726977 0.01732903
## [61] 0.01740587 0.01752546 0.01765476 0.01778855 0.01791941 0.01797282
## [67] 0.01801804 0.01806868 0.01812816 0.01820979 0.01832234 0.01842772
## [73] 0.01852838 0.01864740 0.01877048 0.01891523 0.01909299 0.01929420
## [79] 0.01959339 0.01993123 0.02028234 0.02064877 0.02100748 0.02135643
## [85] 0.02170382 0.02203093 0.02233807 0.02263722 0.02289200
##
## $cvlo
##  [1] 0.15149171 0.13528048 0.12121342 0.10859868 0.09642793 0.08614759
##  [7] 0.07705556 0.06924342 0.06234969 0.05621786 0.05103897 0.04672009
## [13] 0.04305195 0.03979195 0.03696681 0.03455074 0.03241965 0.03051958
## [19] 0.02866450 0.02699929 0.02558026 0.02439422 0.02342713 0.02259299
## [25] 0.02191136 0.02135441 0.02080305 0.02020834 0.01957825 0.01892316
## [31] 0.01831301 0.01775341 0.01724497 0.01680380 0.01641563 0.01607009
## [37] 0.01574650 0.01547096 0.01524479 0.01503397 0.01484467 0.01468496
## [43] 0.01457318 0.01450549 0.01444386 0.01439171 0.01436233 0.01435763
## [49] 0.01436269 0.01436922 0.01438909 0.01443217 0.01448670 0.01452614
## [55] 0.01451702 0.01449283 0.01447317 0.01447168 0.01448027 0.01451584
## [61] 0.01458405 0.01468361 0.01479310 0.01490878 0.01502345 0.01512955
## [67] 0.01524789 0.01537025 0.01547804 0.01558566 0.01571356 0.01583751
## [73] 0.01595769 0.01609044 0.01622386 0.01637626 0.01654613 0.01671326
## [79] 0.01689033 0.01706106 0.01722508 0.01738863 0.01755094 0.01770712
## [85] 0.01785742 0.01799811 0.01813248 0.01827958 0.01838635
##
## $nzero
##  s0  s1  s2  s3  s4  s5  s6  s7  s8  s9 s10 s11 s12 s13 s14 s15 s16 s17
##   0   1   1   2   2   2   3   3   5   5   5   5   6   6   6   7   7   8
## s18 s19 s20 s21 s22 s23 s24 s25 s26 s27 s28 s29 s30 s31 s32 s33 s34 s35
##  10  11  11  11  12  13  13  13  16  21  23  26  28  32  33  34  39  41
## s36 s37 s38 s39 s40 s41 s42 s43 s44 s45 s46 s47 s48 s49 s50 s51 s52 s53
##  43  45  49  52  55  60  67  72  77  81  91  98 103 108 112 119 127 136
## s54 s55 s56 s57 s58 s59 s60 s61 s62 s63 s64 s65 s66 s67 s68 s69 s70 s71
## 142 146 151 155 159 163 168 170 173 175 178 183 184 183 189 193 198 200
## s72 s73 s74 s75 s76 s77 s78 s79 s80 s81 s82 s83 s84 s85 s86 s87 s88
## 204 205 205 210 210 214 216 218 217 216 218 218 219 222 226 228 228
##
## $name
##                 mse
## "Mean-Squared Error"
##
## $glmnet.fit
##
## Call:  glmnet(x = x, y = y)
##
##         Df   %Dev    Lambda
##   [1,]   0 0.0000 3.231e-01
##   [2,]   1 0.1127 2.944e-01
##   [3,]   1 0.2062 2.682e-01
##   [4,]   2 0.2907 2.444e-01
##   [5,]   2 0.3710 2.227e-01
##   [6,]   2 0.4377 2.029e-01
##   [7,]   3 0.4988 1.849e-01
##   [8,]   3 0.5497 1.684e-01
##   [9,]   5 0.5956 1.535e-01
##  [10,]   5 0.6354 1.398e-01
##  [11,]   5 0.6684 1.274e-01
##  [12,]   5 0.6959 1.161e-01
##  [13,]   6 0.7190 1.058e-01
##  [14,]   6 0.7406 9.639e-02
##  [15,]   6 0.7585 8.783e-02
##  [16,]   7 0.7742 8.002e-02
##  [17,]   7 0.7877 7.292e-02
##  [18,]   8 0.7994 6.644e-02
##  [19,]  10 0.8113 6.054e-02
```

```
##  [20,]  11 0.8230 5.516e-02
##  [21,]  11 0.8327 5.026e-02
##  [22,]  11 0.8407 4.579e-02
##  [23,]  12 0.8479 4.172e-02
##  [24,]  13 0.8542 3.802e-02
##  [25,]  13 0.8597 3.464e-02
##  [26,]  13 0.8642 3.156e-02
##  [27,]  16 0.8685 2.876e-02
##  [28,]  21 0.8750 2.620e-02
##  [29,]  23 0.8811 2.388e-02
##  [30,]  26 0.8868 2.176e-02
##  [31,]  28 0.8919 1.982e-02
##  [32,]  32 0.8969 1.806e-02
##  [33,]  33 0.9012 1.646e-02
##  [34,]  34 0.9048 1.500e-02
##  [35,]  39 0.9083 1.366e-02
##  [36,]  41 0.9115 1.245e-02
##  [37,]  43 0.9143 1.134e-02
##  [38,]  45 0.9169 1.034e-02
##  [39,]  49 0.9193 9.417e-03
##  [40,]  52 0.9218 8.581e-03
##  [41,]  55 0.9241 7.818e-03
##  [42,]  60 0.9261 7.124e-03
##  [43,]  67 0.9282 6.491e-03
##  [44,]  72 0.9301 5.914e-03
##  [45,]  77 0.9319 5.389e-03
##  [46,]  81 0.9335 4.910e-03
##  [47,]  91 0.9351 4.474e-03
##  [48,]  98 0.9367 4.077e-03
##  [49,] 103 0.9382 3.714e-03
##  [50,] 108 0.9399 3.384e-03
##  [51,] 112 0.9412 3.084e-03
##  [52,] 119 0.9424 2.810e-03
##  [53,] 127 0.9435 2.560e-03
##  [54,] 136 0.9447 2.333e-03
##  [55,] 142 0.9459 2.126e-03
##  [56,] 146 0.9470 1.937e-03
##  [57,] 151 0.9479 1.765e-03
##  [58,] 155 0.9488 1.608e-03
##  [59,] 159 0.9495 1.465e-03
##  [60,] 163 0.9502 1.335e-03
##  [61,] 168 0.9508 1.216e-03
##  [62,] 170 0.9514 1.108e-03
##  [63,] 173 0.9520 1.010e-03
##  [64,] 175 0.9529 9.201e-04
##  [65,] 178 0.9537 8.384e-04
##  [66,] 183 0.9543 7.639e-04
##  [67,] 184 0.9549 6.960e-04
##  [68,] 183 0.9555 6.342e-04
##  [69,] 189 0.9559 5.778e-04
##  [70,] 193 0.9563 5.265e-04
##  [71,] 198 0.9567 4.797e-04
##  [72,] 200 0.9570 4.371e-04
##  [73,] 204 0.9573 3.983e-04
##  [74,] 205 0.9576 3.629e-04
##  [75,] 205 0.9578 3.307e-04
##  [76,] 210 0.9580 3.013e-04
##  [77,] 210 0.9581 2.745e-04
##  [78,] 214 0.9583 2.501e-04
##  [79,] 216 0.9584 2.279e-04
##  [80,] 218 0.9585 2.077e-04
##  [81,] 217 0.9586 1.892e-04
##  [82,] 216 0.9587 1.724e-04
##  [83,] 218 0.9588 1.571e-04
##  [84,] 218 0.9588 1.431e-04
##  [85,] 219 0.9589 1.304e-04
```

```
##  [86,] 222 0.9590 1.188e-04
##  [87,] 226 0.9590 1.083e-04
##  [88,] 228 0.9590 9.866e-05
##  [89,] 228 0.9591 8.989e-05
##  [90,] 228 0.9591 8.191e-05
##  [91,] 229 0.9592 7.463e-05
##  [92,] 229 0.9592 6.800e-05
##  [93,] 229 0.9592 6.196e-05
##  [94,] 229 0.9592 5.646e-05
##  [95,] 228 0.9593 5.144e-05
##  [96,] 229 0.9593 4.687e-05
##  [97,] 228 0.9593 4.271e-05
##  [98,] 229 0.9594 3.891e-05
##  [99,] 231 0.9594 3.546e-05
## [100,] 230 0.9594 3.231e-05
##
## $lambda.min
## [1] 0.004474026
##
## $lambda.1se
## [1] 0.01134329
##
## attr(,"class")
## [1] "cv.glmnet"
```

```
plot(cv.lasso)
```



The plot indicates the different lambda values that were tried and the mean squared error. (Different between estimator and estimated). The two lines indicate the minimum lambda value and the regularized lambda value which is within the 1 standard error or minimum lambda value.

```
# Optimal Lambda
penalty <- cv.lasso$lambda.1se

# Fit lasso with minimal lambda value
fit1 <-glmnet(x, y, alpha = 1, lambda = penalty)
coef(fit1)
```

```
## 262 x 1 sparse Matrix of class "dgCMatrix"
##                                 s0
## (Intercept)           4.676284e+00
## (Intercept)           .
## MSSubClass           -6.499003e-05
## MSZoningFV            .
## MSZoningRH            .
## MSZoningRL            .
## MSZoningRM           -5.048398e-02
## LotFrontage           2.521740e-04
## LotArea               1.943543e-06
## StreetPave            .
## AlleyNone             .
## AlleyPave             .
## LotShapeIR2           .
## LotShapeIR3           .
## LotShapeReg           .
## LandContourHLS        .
## LandContourLow        .
## LandContourLvl        .
## UtilitiesNoSeWa       .
## LotConfigCulDSac      .
## LotConfigFR2          .
## LotConfigFR3          .
## LotConfigInside       .
## LandSlopeMod          .
## LandSlopeSev          .
## NeighborhoodBlueste   .
## NeighborhoodBrDale    .
## NeighborhoodBrkSide   .
## NeighborhoodClearCr   .
## NeighborhoodCollgCr   .
## NeighborhoodCrawfor   6.888189e-02
## NeighborhoodEdwards  -1.762436e-02
## NeighborhoodGilbert   .
## NeighborhoodIDOTRR    .
## NeighborhoodMeadowV  -7.300101e-03
## NeighborhoodMitchel   .
## NeighborhoodNAmes     .
## NeighborhoodNoRidge   .
## NeighborhoodNPkVill   .
## NeighborhoodNridgHt   1.405692e-02
## NeighborhoodNWAmes    .
## NeighborhoodOldTown   .
## NeighborhoodSawyer    .
## NeighborhoodSawyerW   .
## NeighborhoodSomerst   .
## NeighborhoodStoneBr   6.744703e-02
## NeighborhoodSWISU     .
## NeighborhoodTimber    .
## NeighborhoodVeenker   .
## Condition1Feedr       .
## Condition1Norm        .
## Condition1PosA        .
## Condition1PosN        .
## Condition1RRAe       -2.193028e-02
## Condition1RRAn        .
## Condition1RRNe        .
## Condition1RRNn        .
## Condition2Feedr       .
## Condition2Norm        .
## Condition2PosA        .
## Condition2PosN        .
## Condition2RRAe        .
## Condition2RRAn        .
```

```
## Condition2RRNn        .
## BldgType2fmCon        .
## BldgTypeDuplex        .
## BldgTypeTwnhs        -3.779040e-02
## BldgTypeTwnhsE        .
## HouseStyle1.5Unf      .
## HouseStyle1Story      .
## HouseStyle2.5Fin      .
## HouseStyle2.5Unf      .
## HouseStyle2Story      .
## HouseStyleSFoyer      .
## HouseStyleSLvl        .
## OverallQual           6.510139e-02
## OverallCond           2.670802e-02
## YearBuilt             1.786329e-03
## YearRemodAdd          1.312357e-03
## RoofStyleGable        .
## RoofStyleGambrel      .
## RoofStyleHip          .
## RoofStyleMansard      .
## RoofStyleShed         .
## RoofMatlCompShg       .
## RoofMatlMembran       .
## RoofMatlMetal         .
## RoofMatlRoll          .
## RoofMatlTar&Grv       .
## RoofMatlWdShake       .
## RoofMatlWdShngl       .
## Exterior1stAsphShn    .
## Exterior1stBrkComm   -3.571720e-01
## Exterior1stBrkFace    .
## Exterior1stCBlock     .
## Exterior1stCemntBd    .
## Exterior1stHdBoard    .
## Exterior1stImStucc    .
## Exterior1stMetalSd    .
## Exterior1stPlywood    .
## Exterior1stStone      .
## Exterior1stStucco     .
## Exterior1stVinylSd    .
## Exterior1stWd Sdng    .
## Exterior1stWdShing    .
## Exterior2ndAsphShn    .
## Exterior2ndBrk Cmn    .
## Exterior2ndBrkFace    .
## Exterior2ndCBlock     .
## Exterior2ndCmentBd    .
## Exterior2ndHdBoard    .
## Exterior2ndImStucc    .
## Exterior2ndMetalSd    .
## Exterior2ndOther      .
## Exterior2ndPlywood    .
## Exterior2ndStone      .
## Exterior2ndStucco     .
## Exterior2ndVinylSd    .
## Exterior2ndWd Sdng    .
## Exterior2ndWd Shng    .
## MasVnrTypeBrkFace     .
## MasVnrTypeNone        .
## MasVnrTypeStone       7.727898e-03
## MasVnrArea            .
## ExterQualFa           .
## ExterQualGd           .
## ExterQualTA           .
## ExterCondFa          -6.194946e-04
## ExterCondGd           .
```

```
## ExterCondPo           .
## ExterCondTA           .
## FoundationCBlock      .
## FoundationPConc       1.721587e-02
## FoundationSlab        .
## FoundationStone       .
## FoundationWood        .
## BsmtQualFa            .
## BsmtQualGd            .
## BsmtQualNone          .
## BsmtQualTA            .
## BsmtCondGd            .
## BsmtCondNone          .
## BsmtCondPo            .
## BsmtCondTA            .
## BsmtExposureGd        4.011704e-02
## BsmtExposureMn        .
## BsmtExposureNo       -7.462634e-03
## BsmtExposureNone      .
## BsmtFinType1BLQ       .
## BsmtFinType1GLQ       5.214068e-03
## BsmtFinType1LwQ       .
## BsmtFinType1None      .
## BsmtFinType1Rec       .
## BsmtFinType1Unf       .
## BsmtFinSF1            6.040894e-05
## BsmtFinType2BLQ       .
## BsmtFinType2GLQ       .
## BsmtFinType2LwQ       .
## BsmtFinType2None      .
## BsmtFinType2Rec       .
## BsmtFinType2Unf       .
## BsmtFinSF2            .
## BsmtUnfSF             .
## TotalBsmtSF           1.093849e-04
## HeatingGasA           .
## HeatingGasW           .
## HeatingGrav           .
## HeatingOthW           .
## HeatingWall           .
## HeatingQCFa           .
## HeatingQCGd           .
## HeatingQCPo           .
## HeatingQCTA          -1.004414e-02
## CentralAirY           3.193076e-02
## ElectricalFuseF       .
## ElectricalFuseP       .
## ElectricalMix         .
## ElectricalNone        .
## ElectricalSBrkr       .
## X1stFlrSF             .
## X2ndFlrSF             .
## LowQualFinSF         -4.625068e-05
## GrLivArea             2.650145e-04
## BsmtFullBath          1.438722e-02
## BsmtHalfBath          .
## FullBath              .
## HalfBath              5.997652e-03
## BedroomAbvGr          .
## KitchenAbvGr         -6.818221e-02
## KitchenQualFa         .
## KitchenQualGd         .
## KitchenQualTA         .
## TotRmsAbvGrd          .
## FunctionalMaj2       -4.546045e-02
## FunctionalMin1        .
```

```
## FunctionalMin2        .
## FunctionalMod         .
## FunctionalSev         .
## FunctionalTyp         4.107992e-03
## Fireplaces            7.188212e-03
## FireplaceQuFa         .
## FireplaceQuGd         1.430893e-02
## FireplaceQuNone      -2.123332e-02
## FireplaceQuPo         .
## FireplaceQuTA         .
## GarageTypeAttchd      .
## GarageTypeBasment     .
## GarageTypeBuiltIn     .
## GarageTypeCarPort     .
## GarageTypeDetchd      .
## GarageTypeNone        .
## GarageYrBlt           9.822005e-06
## GarageFinishNone      .
## GarageFinishRFn       .
## GarageFinishUnf       .
## GarageCars            3.092636e-02
## GarageArea            9.460607e-05
## GarageQualFa          .
## GarageQualGd          .
## GarageQualNone        .
## GarageQualPo          .
## GarageQualTA          1.441533e-02
## GarageCondFa          .
## GarageCondGd          .
## GarageCondNone        .
## GarageCondPo          .
## GarageCondTA          .
## PavedDriveP           .
## PavedDriveY           .
## WoodDeckSF            1.826999e-05
## OpenPorchSF           .
## EnclosedPorch         .
## X3SsnPorch            .
## ScreenPorch           .
## PoolArea              .
## PoolQCFa              .
## PoolQCGd              .
## PoolQCNone            .
## FenceGdWo             .
## FenceMnPrv            .
## FenceMnWw             .
## FenceNone             .
## MiscFeatureNone       .
## MiscFeatureOthr       .
## MiscFeatureShed       .
## MiscFeatureTenC       .
## MiscVal               .
## MoSold                .
## YrSold                .
## SaleTypeCon           .
## SaleTypeConLD         .
## SaleTypeConLI         .
## SaleTypeConLw         .
## SaleTypeCWD           .
## SaleTypeNew           3.501999e-02
## SaleTypeOth           .
## SaleTypeWD            .
## SaleConditionAdjLand  .
## SaleConditionAlloca   .
## SaleConditionFamily  -8.927139e-02
```

```
## SaleConditionNormal   .
## SaleConditionPartial  .
```

```
varImp(fit1, lambda = cv.lasso$lambda.1se)
```

```
##          Overall
## 1   4.676284e+00
## 2   0.000000e+00
## 3   6.499003e-05
## 4   0.000000e+00
## 5   0.000000e+00
## 6   0.000000e+00
## 7   5.048398e-02
## 8   2.521740e-04
## 9   1.943543e-06
## 10  0.000000e+00
## 11  0.000000e+00
## 12  0.000000e+00
## 13  0.000000e+00
## 14  0.000000e+00
## 15  0.000000e+00
## 16  0.000000e+00
## 17  0.000000e+00
## 18  0.000000e+00
## 19  0.000000e+00
## 20  0.000000e+00
## 21  0.000000e+00
## 22  0.000000e+00
## 23  0.000000e+00
## 24  0.000000e+00
## 25  0.000000e+00
## 26  0.000000e+00
## 27  0.000000e+00
## 28  0.000000e+00
## 29  0.000000e+00
## 30  0.000000e+00
## 31  6.888189e-02
## 32  1.762436e-02
## 33  0.000000e+00
## 34  0.000000e+00
## 35  7.300101e-03
## 36  0.000000e+00
## 37  0.000000e+00
## 38  0.000000e+00
## 39  0.000000e+00
## 40  1.405692e-02
## 41  0.000000e+00
## 42  0.000000e+00
## 43  0.000000e+00
## 44  0.000000e+00
## 45  0.000000e+00
## 46  6.744703e-02
## 47  0.000000e+00
## 48  0.000000e+00
## 49  0.000000e+00
## 50  0.000000e+00
## 51  0.000000e+00
## 52  0.000000e+00
## 53  0.000000e+00
## 54  2.193028e-02
## 55  0.000000e+00
## 56  0.000000e+00
## 57  0.000000e+00
## 58  0.000000e+00
## 59  0.000000e+00
## 60  0.000000e+00
## 61  0.000000e+00
## 62  0.000000e+00
## 63  0.000000e+00
## 64  0.000000e+00
```

```
## 65  0.000000e+00
## 66  0.000000e+00
## 67  3.779040e-02
## 68  0.000000e+00
## 69  0.000000e+00
## 70  0.000000e+00
## 71  0.000000e+00
## 72  0.000000e+00
## 73  0.000000e+00
## 74  0.000000e+00
## 75  0.000000e+00
## 76  6.510139e-02
## 77  2.670802e-02
## 78  1.786329e-03
## 79  1.312357e-03
## 80  0.000000e+00
## 81  0.000000e+00
## 82  0.000000e+00
## 83  0.000000e+00
## 84  0.000000e+00
## 85  0.000000e+00
## 86  0.000000e+00
## 87  0.000000e+00
## 88  0.000000e+00
## 89  0.000000e+00
## 90  0.000000e+00
## 91  0.000000e+00
## 92  0.000000e+00
## 93  3.571720e-01
## 94  0.000000e+00
## 95  0.000000e+00
## 96  0.000000e+00
## 97  0.000000e+00
## 98  0.000000e+00
## 99  0.000000e+00
## 100 0.000000e+00
## 101 0.000000e+00
## 102 0.000000e+00
## 103 0.000000e+00
## 104 0.000000e+00
## 105 0.000000e+00
## 106 0.000000e+00
## 107 0.000000e+00
## 108 0.000000e+00
## 109 0.000000e+00
## 110 0.000000e+00
## 111 0.000000e+00
## 112 0.000000e+00
## 113 0.000000e+00
## 114 0.000000e+00
## 115 0.000000e+00
## 116 0.000000e+00
## 117 0.000000e+00
## 118 0.000000e+00
## 119 0.000000e+00
## 120 0.000000e+00
## 121 0.000000e+00
## 122 0.000000e+00
## 123 7.727898e-03
## 124 0.000000e+00
## 125 0.000000e+00
## 126 0.000000e+00
## 127 0.000000e+00
## 128 6.194946e-04
## 129 0.000000e+00
## 130 0.000000e+00
```

```
## 131 0.000000e+00
## 132 0.000000e+00
## 133 1.721587e-02
## 134 0.000000e+00
## 135 0.000000e+00
## 136 0.000000e+00
## 137 0.000000e+00
## 138 0.000000e+00
## 139 0.000000e+00
## 140 0.000000e+00
## 141 0.000000e+00
## 142 0.000000e+00
## 143 0.000000e+00
## 144 0.000000e+00
## 145 4.011704e-02
## 146 0.000000e+00
## 147 7.462634e-03
## 148 0.000000e+00
## 149 0.000000e+00
## 150 5.214068e-03
## 151 0.000000e+00
## 152 0.000000e+00
## 153 0.000000e+00
## 154 0.000000e+00
## 155 6.040894e-05
## 156 0.000000e+00
## 157 0.000000e+00
## 158 0.000000e+00
## 159 0.000000e+00
## 160 0.000000e+00
## 161 0.000000e+00
## 162 0.000000e+00
## 163 0.000000e+00
## 164 1.093849e-04
## 165 0.000000e+00
## 166 0.000000e+00
## 167 0.000000e+00
## 168 0.000000e+00
## 169 0.000000e+00
## 170 0.000000e+00
## 171 0.000000e+00
## 172 0.000000e+00
## 173 1.004414e-02
## 174 3.193076e-02
## 175 0.000000e+00
## 176 0.000000e+00
## 177 0.000000e+00
## 178 0.000000e+00
## 179 0.000000e+00
## 180 0.000000e+00
## 181 0.000000e+00
## 182 4.625068e-05
## 183 2.650145e-04
## 184 1.438722e-02
## 185 0.000000e+00
## 186 0.000000e+00
## 187 5.997652e-03
## 188 0.000000e+00
## 189 6.818221e-02
## 190 0.000000e+00
## 191 0.000000e+00
## 192 0.000000e+00
## 193 0.000000e+00
## 194 4.546045e-02
## 195 0.000000e+00
## 196 0.000000e+00
```

```
## 197 0.000000e+00
## 198 0.000000e+00
## 199 4.107992e-03
## 200 7.188212e-03
## 201 0.000000e+00
## 202 1.430893e-02
## 203 2.123332e-02
## 204 0.000000e+00
## 205 0.000000e+00
## 206 0.000000e+00
## 207 0.000000e+00
## 208 0.000000e+00
## 209 0.000000e+00
## 210 0.000000e+00
## 211 0.000000e+00
## 212 9.822005e-06
## 213 0.000000e+00
## 214 0.000000e+00
## 215 0.000000e+00
## 216 3.092636e-02
## 217 9.460607e-05
## 218 0.000000e+00
## 219 0.000000e+00
## 220 0.000000e+00
## 221 0.000000e+00
## 222 1.441533e-02
## 223 0.000000e+00
## 224 0.000000e+00
## 225 0.000000e+00
## 226 0.000000e+00
## 227 0.000000e+00
## 228 0.000000e+00
## 229 0.000000e+00
## 230 1.826999e-05
## 231 0.000000e+00
## 232 0.000000e+00
## 233 0.000000e+00
## 234 0.000000e+00
## 235 0.000000e+00
## 236 0.000000e+00
## 237 0.000000e+00
## 238 0.000000e+00
## 239 0.000000e+00
## 240 0.000000e+00
## 241 0.000000e+00
## 242 0.000000e+00
## 243 0.000000e+00
## 244 0.000000e+00
## 245 0.000000e+00
## 246 0.000000e+00
## 247 0.000000e+00
## 248 0.000000e+00
## 249 0.000000e+00
## 250 0.000000e+00
## 251 0.000000e+00
## 252 0.000000e+00
## 253 0.000000e+00
## 254 0.000000e+00
## 255 3.501999e-02
## 256 0.000000e+00
## 257 0.000000e+00
## 258 0.000000e+00
## 259 0.000000e+00
## 260 8.927139e-02
## 261 0.000000e+00
## 262 0.000000e+00
```

Make Predictions using the Lambda values on test data

```
# First convert test data into a matrix
test1 = testing1[,2:81]

test.x = model.matrix(SalePrice~., data = test1)
#dim(test.x)

results <-predict(fit1, newx = test.x, s=penalty, type="response")

# summarize accuracy
test1$SalePrice = log(test1$SalePrice + 1)
mse <- mean(results - test1$SalePrice)^2

# Prediction error
prediction.error <- (mse*100)/mean(train$SalePrice)
prediction.error
```

```
## [1] 745.4876
```

So, the percentage error is 65%.

Random Forest Regression model.

http://stackoverflow.com/questions/32014311/pca-for-dimensionality-reduction-before-random-forest (http://stackoverflow.com/questions/32014311/pca-for-dimensionality-reduction-before-random-forest)

```
require(randomForest)

rf.model <- randomForest(SalePrice ~ ., data = train, ntree = 500, replace = TRUE)
summary(rf.model)
```

```
##                 Length Class  Mode
## call               5   -none- call
## type               1   -none- character
## predicted       1460   -none- numeric
## mse              500   -none- numeric
## rsq              500   -none- numeric
## oob.times       1460   -none- numeric
## importance        81   -none- numeric
## importanceSD       0   -none- NULL
## localImportance    0   -none- NULL
## proximity          0   -none- NULL
## ntree              1   -none- numeric
## mtry               1   -none- numeric
## forest            11   -none- list
## coefs              0   -none- NULL
## y               1460   -none- numeric
## test               0   -none- NULL
## inbag              0   -none- NULL
## terms              3   terms  call
```