Annotated Code and Graphics for the *Ames* Problem Statement

12 March 2018

Exploratory Data Analysis

We begin by loading the packages needed for the analysis.

```
library(tidyverse)
library(rapportools)
library(ggplot2)
library(fastDummies)
library(gridExtra)
library(broom)
library(glmnet)
library(cowplot)
library(gbm)
library(pcaMethods)
library(vegan)
library(cluster)
library(fpc)
library(usdm)
library(DataExplorer)
library(ggbiplot)
library(ggfortify)
```

We download the data and begin prepping it.

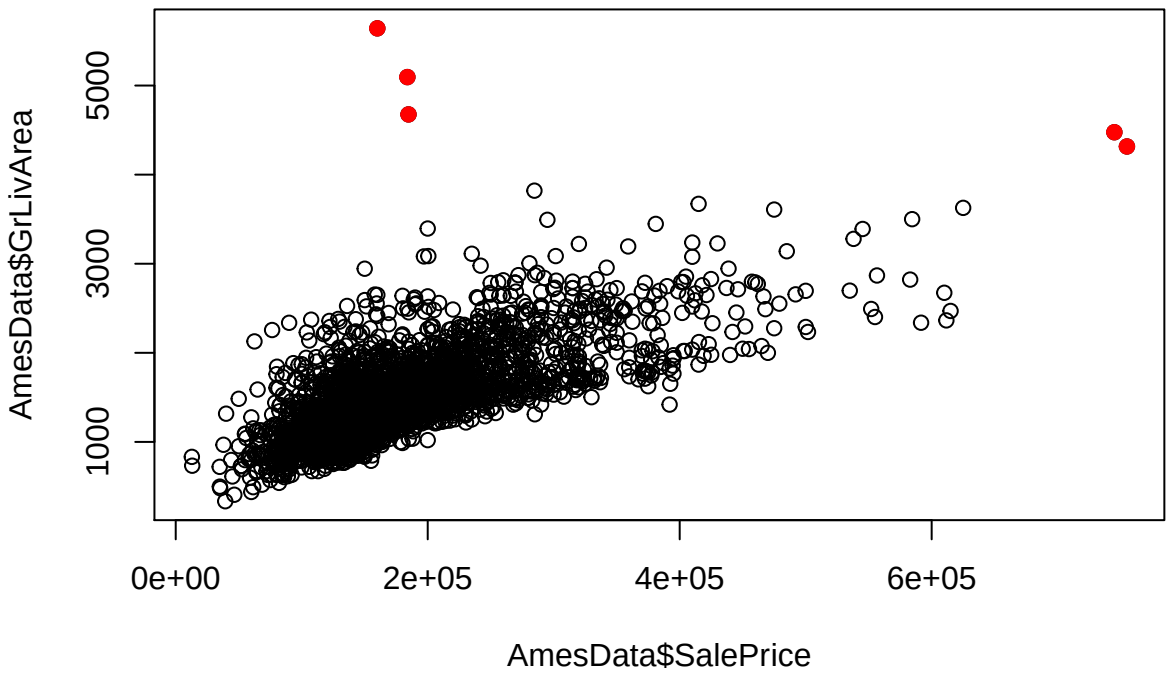
```
AmesURL <- 'https://www.openintro.org/stat/data/ames.csv'
AmesData <- read.csv(url(AmesURL), sep = ",", header = TRUE, stringsAsFactors = FALSE)
```

The column headers include characters that might cause trouble later so we get rid of them.

```
names(AmesData) <- gsub('\\.', '', names(AmesData))
```

Let's make a visual of the data and remove some outliers, as the data set's creator advises. First, identify the outliers in the original data. They are properties over 4000 feet2 in the `GrLivArea` feature. There are five according to the literature.

```
plot(AmesData$SalePrice, AmesData$GrLivArea)
points(AmesData$SalePrice[c(1499, 2181, 2182, 1761, 1768)], AmesData$GrLivArea[c(1499, 2181, 2182, 1761
```



Sort the data by `GrLivArea` to identify the top five houses.

```
sorted <- AmesData[order(AmesData$GrLivArea, decreasing = TRUE), ]
head(sorted)
```

```
##      Order       PID MSSubClass MSZoning LotFrontage LotArea Street Alley
## 1499  1499 908154235         60       RL         313   63887   Pave  <NA>
## 2181  2181 908154195         20       RL         128   39290   Pave  <NA>
## 2182  2182 908154205         60       RL         130   40094   Pave  <NA>
## 1761  1761 528320050         60       RL         160   15623   Pave  <NA>
## 1768  1768 528351010         60       RL         104   21535   Pave  <NA>
## 1498  1498 908154080         20       RL         123   47007   Pave  <NA>
##      LotShape LandContour Utilities LotConfig LandSlope Neighborhood
## 1499      IR3         Bnk    AllPub    Corner       Gtl      Edwards
## 2181      IR1         Bnk    AllPub    Inside       Gtl      Edwards
## 2182      IR1         Bnk    AllPub    Inside       Gtl      Edwards
## 1761      IR1         Lvl    AllPub    Corner       Gtl      NoRidge
## 1768      IR1         Lvl    AllPub    Corner       Gtl      NoRidge
## 1498      IR1         Lvl    AllPub    Inside       Gtl      Edwards
##      Condition1 Condition2 BldgType HouseStyle OverallQual OverallCond
## 1499      Feedr       Norm     1Fam     2Story          10           5
## 2181       Norm       Norm     1Fam     1Story          10           5
## 2182       PosN       PosN     1Fam     2Story          10           5
## 1761       Norm       Norm     1Fam     2Story          10           5
## 1768       Norm       Norm     1Fam     2Story          10           6
## 1498       Norm       Norm     1Fam     1Story           5           7
##      YearBuilt YearRemodAdd RoofStyle RoofMatl Exterior1st Exterior2nd
```

```
## 1499      2008         2008       Hip  ClyTile      Stucco      Stucco
## 2181      2008         2009       Hip  CompShg     CemntBd     CmentBd
## 2182      2007         2008       Hip  CompShg     CemntBd     CmentBd
## 1761      1996         1996       Hip  CompShg     Wd Sdng     ImStucc
## 1768      1994         1995     Gable  WdShngl     HdBoard     HdBoard
## 1498      1959         1996     Gable  CompShg     Plywood     Plywood
##      MasVnrType MasVnrArea ExterQual ExterCond Foundation BsmtQual
## 1499      Stone        796        Ex        TA      PConc       Ex
## 2181      Stone       1224        Ex        TA      PConc       Ex
## 2182      Stone        762        Ex        TA      PConc       Ex
## 1761       None          0        Gd        TA      PConc       Ex
## 1768    BrkFace       1170        Ex        TA      PConc       Ex
## 1498       None          0        TA        TA       Slab     <NA>
##      BsmtCond BsmtExposure BsmtFinType1 BsmtFinSF1 BsmtFinType2 BsmtFinSF2
## 1499       TA           Gd          GLQ       5644          Unf          0
## 2181       TA           Gd          GLQ       4010          Unf          0
## 2182       TA           Gd          GLQ       2260          Unf          0
## 1761       TA           Av          GLQ       2096          Unf          0
## 1768       TA           Gd          GLQ       1455          Unf          0
## 1498     <NA>         <NA>         <NA>          0         <NA>          0
##      BsmtUnfSF TotalBsmtSF Heating HeatingQC CentralAir Electrical
## 1499       466        6110    GasA        Ex          Y      SBrkr
## 2181      1085        5095    GasA        Ex          Y      SBrkr
## 2182       878        3138    GasA        Ex          Y      SBrkr
## 1761       300        2396    GasA        Ex          Y      SBrkr
## 1768       989        2444    GasA        Ex          Y      SBrkr
## 1498         0           0    GasA        TA          Y      SBrkr
##      X1stFlrSF X2ndFlrSF LowQualFinSF GrLivArea BsmtFullBath BsmtHalfBath
## 1499      4692       950            0      5642            2            0
## 2181      5095         0            0      5095            1            1
## 2182      3138      1538            0      4676            1            0
## 1761      2411      2065            0      4476            1            0
## 1768      2444      1872            0      4316            0            1
## 1498      3820         0            0      3820           NA           NA
##      FullBath HalfBath BedroomAbvGr KitchenAbvGr KitchenQual TotRmsAbvGrd
## 1499        2        1            3            1          Ex           12
## 2181        2        1            2            1          Ex           15
## 2182        3        1            3            1          Ex           11
## 1761        3        1            4            1          Ex           10
## 1768        3        1            4            1          Ex           10
## 1498        3        1            5            1          Ex           11
##      Functional Fireplaces FireplaceQu GarageType GarageYrBlt GarageFinish
## 1499        Typ          3          Gd     Attchd        2008          Fin
## 2181        Typ          2          Gd     Attchd        2008          Fin
## 2182        Typ          1          Gd    BuiltIn        2007          Fin
## 1761        Typ          2          TA     Attchd        1996          Fin
## 1768        Typ          2          Ex     Attchd        1994          Fin
## 1498        Typ          2          Gd     Attchd        1959          Unf
##      GarageCars GarageArea GarageQual GarageCond PavedDrive WoodDeckSF
## 1499          2       1418         TA         TA          Y        214
## 2181          3       1154         TA         TA          Y        546
## 2182          3        884         TA         TA          Y        208
## 1761          3        813         TA         TA          Y        171
## 1768          3        832         TA         TA          Y        382
```

```
## 1498          2        624         TA         TA          Y          0
##      OpenPorchSF EnclosedPorch X3SsnPorch ScreenPorch PoolArea PoolQC
## 1499         292             0          0           0      480     Gd
## 2181         484             0          0           0        0   <NA>
## 2182         406             0          0           0        0   <NA>
## 1761          78             0          0           0      555     Ex
## 1768          50             0          0           0        0   <NA>
## 1498         372             0          0           0        0   <NA>
##      Fence MiscFeature MiscVal MoSold YrSold SaleType SaleCondition
## 1499  <NA>        <NA>       0      1   2008      New       Partial
## 2181  <NA>        Elev   17000     10   2007      New       Partial
## 2182  <NA>        <NA>       0     10   2007      New       Partial
## 1761 MnPrv        <NA>       0      7   2007      WD        Abnorml
## 1768  <NA>        <NA>       0      1   2007      WD         Normal
## 1498  <NA>        <NA>       0      7   2008      WD         Normal
##      SalePrice
## 1499    160000
## 2181    183850
## 2182    184750
## 1761    745000
## 1768    755000
## 1498    284700
```

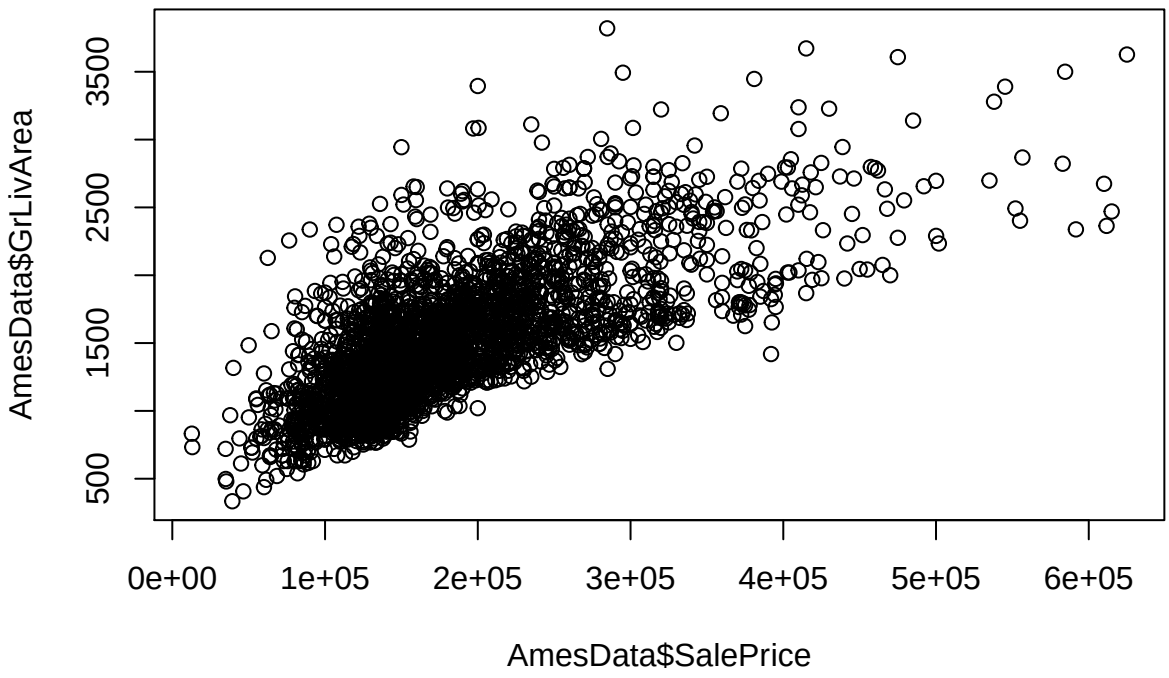
Check data dimensions.

```
dim(AmesData)
```

```
## [1] 2930   82
```

Now remove the outliers and visualize the data to see if the operation is complete.

```
AmesData <- AmesData[-c(1499, 2181, 2182, 1761, 1768), ]
plot(AmesData$SalePrice, AmesData$GrLivArea)
```



Run a sanity check on dimensions just to make sure.

```
dim(AmesData)
```

```
## [1] 2925   82
```

Check to see the memory footprint of the data file.

```
pryr::object_size(AmesData)
```

```
## 1.49 MB
```

Look for any duplicates in the data, then run a summary.

```
anyDuplicated(AmesData)
```

```
## [1] 0
```

```
summary(AmesData)
```

```
##      Order            PID              MSSubClass      MSZoning
##  Min.   :   1   Min.   :5.263e+08   Min.   : 20.0   Length:2925
##  1st Qu.: 732   1st Qu.:5.285e+08   1st Qu.: 20.0   Class :character
##  Median :1463   Median :5.355e+08   Median : 50.0   Mode  :character
##  Mean   :1465   Mean   :7.144e+08   Mean   : 57.4
##  3rd Qu.:2199   3rd Qu.:9.072e+08   3rd Qu.: 70.0
##  Max.   :2930   Max.   :1.007e+09   Max.   :190.0
##
##   LotFrontage        LotArea          Street             Alley
##  Min.   : 21.00   Min.   :  1300   Length:2925        Length:2925
```

```
##  1st Qu.: 58.00   1st Qu.:  7438   Class :character   Class :character
##  Median : 68.00   Median :  9428   Mode  :character   Mode  :character
##  Mean   : 69.02   Mean   : 10104
##  3rd Qu.: 80.00   3rd Qu.: 11515
##  Max.   :313.00   Max.   :215245
##  NA's   :490
##    LotShape         LandContour         Utilities
##  Length:2925        Length:2925        Length:2925
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   LotConfig          LandSlope         Neighborhood
##  Length:2925        Length:2925        Length:2925
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   Condition1         Condition2          BldgType
##  Length:2925        Length:2925        Length:2925
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   HouseStyle         OverallQual      OverallCond      YearBuilt
##  Length:2925        Min.   : 1.000   Min.   :1.000   Min.   :1872
##  Class :character   1st Qu.: 5.000   1st Qu.:5.000   1st Qu.:1954
##  Mode  :character   Median : 6.000   Median :5.000   Median :1973
##                     Mean   : 6.088   Mean   :5.564   Mean   :1971
##                     3rd Qu.: 7.000   3rd Qu.:6.000   3rd Qu.:2001
##                     Max.   :10.000   Max.   :9.000   Max.   :2010
##
##   YearRemodAdd   RoofStyle           RoofMatl         Exterior1st
##  Min.   :1950   Length:2925        Length:2925        Length:2925
##  1st Qu.:1965   Class :character   Class :character   Class :character
##  Median :1993   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1984
##  3rd Qu.:2004
##  Max.   :2010
##
##  Exterior2nd         MasVnrType          MasVnrArea      ExterQual
##  Length:2925        Length:2925        Min.   :   0.0   Length:2925
##  Class :character   Class :character   1st Qu.:   0.0   Class :character
##  Mode  :character   Mode  :character   Median :   0.0   Mode  :character
##                                        Mean   : 100.7
##                                        3rd Qu.: 164.0
##                                        Max.   :1600.0
##                                        NA's   :23
```

```
##   ExterCond          Foundation          BsmtQual
##  Length:2925        Length:2925        Length:2925
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    BsmtCond         BsmtExposure       BsmtFinType1         BsmtFinSF1
##  Length:2925        Length:2925        Length:2925        Min.   :   0.0
##  Class :character   Class :character   Class :character   1st Qu.:   0.0
##  Mode  :character   Mode  :character   Mode  :character   Median : 369.5
##                                                           Mean   : 438.1
##                                                           3rd Qu.: 733.2
##                                                           Max.   :2288.0
##                                                           NA's   :1
##  BsmtFinType2         BsmtFinSF2        BsmtUnfSF       TotalBsmtSF
##  Length:2925        Min.   :   0.00   Min.   :   0.0   Min.   :   0.0
##  Class :character   1st Qu.:   0.00   1st Qu.: 218.8   1st Qu.: 792.8
##  Mode  :character   Median :   0.00   Median : 464.5   Median : 989.5
##                     Mean   :  49.81   Mean   : 558.9   Mean   :1046.9
##                     3rd Qu.:   0.00   3rd Qu.: 801.0   3rd Qu.:1299.2
##                     Max.   :1526.00   Max.   :2336.0   Max.   :3206.0
##                     NA's   :1         NA's   :1        NA's   :1
##    Heating           HeatingQC          CentralAir
##  Length:2925        Length:2925        Length:2925
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   Electrical          X1stFlrSF      X2ndFlrSF       LowQualFinSF
##  Length:2925        Min.   : 334   Min.   :   0.0   Min.   :   0.000
##  Class :character   1st Qu.: 876   1st Qu.:   0.0   1st Qu.:   0.000
##  Mode  :character   Median :1082   Median :   0.0   Median :   0.000
##                     Mean   :1155   Mean   : 333.8   Mean   :   4.685
##                     3rd Qu.:1383   3rd Qu.: 702.0   3rd Qu.:   0.000
##                     Max.   :3820   Max.   :1862.0   Max.   :1064.000
##
##    GrLivArea     BsmtFullBath     BsmtHalfBath        FullBath
##  Min.   : 334   Min.   :0.0000   Min.   :0.00000   Min.   :0.000
##  1st Qu.:1126   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:1.000
##  Median :1441   Median :0.0000   Median :0.00000   Median :2.000
##  Mean   :1494   Mean   :0.4304   Mean   :0.06055   Mean   :1.565
##  3rd Qu.:1740   3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:2.000
##  Max.   :3820   Max.   :3.0000   Max.   :2.00000   Max.   :4.000
##                 NA's   :2        NA's   :2
##     HalfBath       BedroomAbvGr    KitchenAbvGr   KitchenQual
##  Min.   :0.0000   Min.   :0.000   Min.   :0.000   Length:2925
##  1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:1.000   Class :character
##  Median :0.0000   Median :3.000   Median :1.000   Mode  :character
##  Mean   :0.3785   Mean   :2.854   Mean   :1.044
##  3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:1.000
```

```
##  Max.   :2.0000   Max.   :8.000   Max.   :3.000
##
##   TotRmsAbvGrd     Functional          Fireplaces     FireplaceQu
##  Min.   : 2.000   Length:2925        Min.   :0.0000   Length:2925
##  1st Qu.: 5.000   Class :character   1st Qu.:0.0000   Class :character
##  Median : 6.000   Mode  :character   Median :1.0000   Mode  :character
##  Mean   : 6.434                      Mean   :0.5969
##  3rd Qu.: 7.000                      3rd Qu.:1.0000
##  Max.   :14.000                      Max.   :4.0000
##
##   GarageType         GarageYrBlt   GarageFinish         GarageCars
##  Length:2925        Min.   :1895   Length:2925        Min.   :0.000
##  Class :character   1st Qu.:1960   Class :character   1st Qu.:1.000
##  Mode  :character   Median :1979   Mode  :character   Median :2.000
##                     Mean   :1978                      Mean   :1.765
##                     3rd Qu.:2002                      3rd Qu.:2.000
##                     Max.   :2207                      Max.   :5.000
##                     NA's   :159                       NA's   :1
##    GarageArea      GarageQual         GarageCond         PavedDrive
##  Min.   :   0.0   Length:2925        Length:2925        Length:2925
##  1st Qu.: 320.0   Class :character   Class :character   Class :character
##  Median : 480.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 471.9
##  3rd Qu.: 576.0
##  Max.   :1488.0
##  NA's   :1
##    WoodDeckSF       OpenPorchSF     EnclosedPorch       X3SsnPorch
##  Min.   :   0.00   Min.   :  0.00   Min.   :   0.00   Min.   :  0.000
##  1st Qu.:   0.00   1st Qu.:  0.00   1st Qu.:   0.00   1st Qu.:  0.000
##  Median :   0.00   Median : 27.00   Median :   0.00   Median :  0.000
##  Mean   :  93.39   Mean   : 47.17   Mean   :  23.05   Mean   :  2.597
##  3rd Qu.: 168.00   3rd Qu.: 70.00   3rd Qu.:   0.00   3rd Qu.:  0.000
##  Max.   :1424.00   Max.   :742.00   Max.   :1012.00   Max.   :508.000
##
##   ScreenPorch        PoolArea          PoolQC             Fence
##  Min.   :  0.00   Min.   :  0.000   Length:2925        Length:2925
##  1st Qu.:  0.00   1st Qu.:  0.000   Class :character   Class :character
##  Median :  0.00   Median :  0.000   Mode  :character   Mode  :character
##  Mean   : 16.03   Mean   :  1.893
##  3rd Qu.:  0.00   3rd Qu.:  0.000
##  Max.   :576.00   Max.   :800.000
##
##  MiscFeature          MiscVal             MoSold           YrSold
##  Length:2925        Min.   :    0.00   Min.   : 1.000   Min.   :2006
##  Class :character   1st Qu.:    0.00   1st Qu.: 4.000   1st Qu.:2007
##  Mode  :character   Median :    0.00   Median : 6.000   Median :2008
##                     Mean   :   44.91   Mean   : 6.217   Mean   :2008
##                     3rd Qu.:    0.00   3rd Qu.: 8.000   3rd Qu.:2009
##                     Max.   :15500.00   Max.   :12.000   Max.   :2010
##
##    SaleType         SaleCondition        SalePrice
##  Length:2925        Length:2925        Min.   : 12789
##  Class :character   Class :character   1st Qu.:129500
##  Mode  :character   Mode  :character   Median :160000
```

```
##                                         Mean   :180412
##                                         3rd Qu.:213500
##                                         Max.   :625000
##
```

Check the data structure. Notice it contains character strings and not factors. We specified `stringsAsFactors = FALSE` when we downloaded the file. Characters will be easier to work with, at first.

```
str(AmesData)
```

```
## 'data.frame':    2925 obs. of  82 variables:
##  $ Order        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ PID          : int  526301100 526350040 526351010 526353030 527105010 527105030 527127150 5271450
##  $ MSSubClass   : int  20 20 20 20 60 60 120 120 120 60 ...
##  $ MSZoning     : chr  "RL" "RH" "RL" "RL" ...
##  $ LotFrontage  : int  141 80 81 93 74 78 41 43 39 60 ...
##  $ LotArea      : int  31770 11622 14267 11160 13830 9978 4920 5005 5389 7500 ...
##  $ Street       : chr  "Pave" "Pave" "Pave" "Pave" ...
##  $ Alley        : chr  NA NA NA NA ...
##  $ LotShape     : chr  "IR1" "Reg" "IR1" "Reg" ...
##  $ LandContour  : chr  "Lvl" "Lvl" "Lvl" "Lvl" ...
##  $ Utilities    : chr  "AllPub" "AllPub" "AllPub" "AllPub" ...
##  $ LotConfig    : chr  "Corner" "Inside" "Corner" "Corner" ...
##  $ LandSlope    : chr  "Gtl" "Gtl" "Gtl" "Gtl" ...
##  $ Neighborhood : chr  "NAmes" "NAmes" "NAmes" "NAmes" ...
##  $ Condition1   : chr  "Norm" "Feedr" "Norm" "Norm" ...
##  $ Condition2   : chr  "Norm" "Norm" "Norm" "Norm" ...
##  $ BldgType     : chr  "1Fam" "1Fam" "1Fam" "1Fam" ...
##  $ HouseStyle   : chr  "1Story" "1Story" "1Story" "1Story" ...
##  $ OverallQual  : int  6 5 6 7 5 6 8 8 8 7 ...
##  $ OverallCond  : int  5 6 6 5 5 6 5 5 5 5 ...
##  $ YearBuilt    : int  1960 1961 1958 1968 1997 1998 2001 1992 1995 1999 ...
##  $ YearRemodAdd : int  1960 1961 1958 1968 1998 1998 2001 1992 1996 1999 ...
##  $ RoofStyle    : chr  "Hip" "Gable" "Hip" "Hip" ...
##  $ RoofMatl     : chr  "CompShg" "CompShg" "CompShg" "CompShg" ...
##  $ Exterior1st  : chr  "BrkFace" "VinylSd" "Wd Sdng" "BrkFace" ...
##  $ Exterior2nd  : chr  "Plywood" "VinylSd" "Wd Sdng" "BrkFace" ...
##  $ MasVnrType   : chr  "Stone" "None" "BrkFace" "None" ...
##  $ MasVnrArea   : int  112 0 108 0 0 20 0 0 0 0 ...
##  $ ExterQual    : chr  "TA" "TA" "TA" "Gd" ...
##  $ ExterCond    : chr  "TA" "TA" "TA" "TA" ...
##  $ Foundation   : chr  "CBlock" "CBlock" "CBlock" "CBlock" ...
##  $ BsmtQual     : chr  "TA" "TA" "TA" "TA" ...
##  $ BsmtCond     : chr  "Gd" "TA" "TA" "TA" ...
##  $ BsmtExposure : chr  "Gd" "No" "No" "No" ...
##  $ BsmtFinType1 : chr  "BLQ" "Rec" "ALQ" "ALQ" ...
##  $ BsmtFinSF1   : int  639 468 923 1065 791 602 616 263 1180 0 ...
##  $ BsmtFinType2 : chr  "Unf" "LwQ" "Unf" "Unf" ...
##  $ BsmtFinSF2   : int  0 144 0 0 0 0 0 0 0 0 ...
##  $ BsmtUnfSF    : int  441 270 406 1045 137 324 722 1017 415 994 ...
##  $ TotalBsmtSF  : int  1080 882 1329 2110 928 926 1338 1280 1595 994 ...
##  $ Heating      : chr  "GasA" "GasA" "GasA" "GasA" ...
##  $ HeatingQC    : chr  "Fa" "TA" "TA" "Ex" ...
##  $ CentralAir   : chr  "Y" "Y" "Y" "Y" ...
##  $ Electrical   : chr  "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...
```

```
##  $ X1stFlrSF    : int  1656 896 1329 2110 928 926 1338 1280 1616 1028 ...
##  $ X2ndFlrSF    : int  0 0 0 0 701 678 0 0 0 776 ...
##  $ LowQualFinSF : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea    : int  1656 896 1329 2110 1629 1604 1338 1280 1616 1804 ...
##  $ BsmtFullBath : int  1 0 0 1 0 0 1 0 1 0 ...
##  $ BsmtHalfBath : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ FullBath     : int  1 1 1 2 2 2 2 2 2 2 ...
##  $ HalfBath     : int  0 0 1 1 1 1 0 0 0 1 ...
##  $ BedroomAbvGr : int  3 2 3 3 3 3 2 2 2 3 ...
##  $ KitchenAbvGr : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ KitchenQual  : chr  "TA" "TA" "Gd" "Ex" ...
##  $ TotRmsAbvGrd : int  7 5 6 8 6 7 6 5 5 7 ...
##  $ Functional   : chr  "Typ" "Typ" "Typ" "Typ" ...
##  $ Fireplaces   : int  2 0 0 2 1 1 0 0 1 1 ...
##  $ FireplaceQu  : chr  "Gd" NA NA "TA" ...
##  $ GarageType   : chr  "Attchd" "Attchd" "Attchd" "Attchd" ...
##  $ GarageYrBlt  : int  1960 1961 1958 1968 1997 1998 2001 1992 1995 1999 ...
##  $ GarageFinish : chr  "Fin" "Unf" "Unf" "Fin" ...
##  $ GarageCars   : int  2 1 1 2 2 2 2 2 2 2 ...
##  $ GarageArea   : int  528 730 312 522 482 470 582 506 608 442 ...
##  $ GarageQual   : chr  "TA" "TA" "TA" "TA" ...
##  $ GarageCond   : chr  "TA" "TA" "TA" "TA" ...
##  $ PavedDrive   : chr  "P" "Y" "Y" "Y" ...
##  $ WoodDeckSF   : int  210 140 393 0 212 360 0 0 237 140 ...
##  $ OpenPorchSF  : int  62 0 36 0 34 36 0 82 152 60 ...
##  $ EnclosedPorch: int  0 0 0 0 0 0 170 0 0 0 ...
##  $ X3SsnPorch   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ScreenPorch  : int  0 120 0 0 0 0 0 144 0 0 ...
##  $ PoolArea     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC       : chr  NA NA NA NA ...
##  $ Fence        : chr  NA "MnPrv" NA NA ...
##  $ MiscFeature  : chr  NA NA "Gar2" NA ...
##  $ MiscVal      : int  0 0 12500 0 0 0 0 0 0 0 ...
##  $ MoSold       : int  5 6 6 4 3 6 4 1 3 6 ...
##  $ YrSold       : int  2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
##  $ SaleType     : chr  "WD " "WD " "WD " "WD " ...
##  $ SaleCondition: chr  "Normal" "Normal" "Normal" "Normal" ...
##  $ SalePrice    : int  215000 105000 172000 244000 189900 195500 213500 191500 236500 189000 ...
```

Check to see how many empty cells and how many *NA* values we are dealing with. It is in the tens of thousands.

```
sum(is.na(AmesData))
```

```
## [1] 13944
```

```
sum(is.empty(AmesData))
```

```
## [1] 48760
```

```
nan_sums <- colSums(is.na(AmesData))
nan_sums[nan_sums > 0]
```

```
##  LotFrontage        Alley   MasVnrArea     BsmtQual     BsmtCond
##          490         2727           23           79           79
## BsmtExposure BsmtFinType1   BsmtFinSF1 BsmtFinType2   BsmtFinSF2
##           79           79            1           79            1
```

```
##    BsmtUnfSF  TotalBsmtSF BsmtFullBath BsmtHalfBath  FireplaceQu
##            1            1            2            2         1422
##   GarageType  GarageYrBlt GarageFinish   GarageCars   GarageArea
##          157          159          157            1            1
##   GarageQual   GarageCond       PoolQC        Fence  MiscFeature
##          158          158         2914         2354         2820
```

For the numeric features, we need to remove the *NA*s and replace them with the median value for the column. First, create the objects that contain the median values.

```
medianLotFrontage <- median(AmesData$LotFrontage, na.rm = TRUE)
medianMasVnrArea <- median(AmesData$MasVnrArea, na.rm = TRUE)
medianBsmtFinSF1 <- median(AmesData$BsmtFinSF1, na.rm = TRUE)
medianBsmtFinSF2 <- median(AmesData$BsmtFinSF2, na.rm = TRUE)
medianBsmtUnfSF <- median(AmesData$BsmtUnfSF, na.rm = TRUE)
medianTotalBsmtSF <- median(AmesData$TotalBsmtSF, na.rm = TRUE)
medianGarageYrBlt <- median(AmesData$GarageYrBlt, na.rm = TRUE)
medianGarageArea <- median(AmesData$GarageArea, na.rm = TRUE)
```

Second, replace the *NA*s in those columns with the median values.

```
AmesData$LotFrontage[is.na(AmesData$LotFrontage)] <- medianLotFrontage
AmesData$MasVnrArea[is.na(AmesData$MasVnrArea)] <- medianMasVnrArea
AmesData$BsmtFinSF1[is.na(AmesData$BsmtFinSF1)] <- medianBsmtFinSF1
AmesData$BsmtFinSF2[is.na(AmesData$BsmtFinSF2)] <- medianBsmtFinSF2
AmesData$BsmtUnfSF[is.na(AmesData$BsmtUnfSF)] <- medianBsmtUnfSF
AmesData$TotalBsmtSF[is.na(AmesData$TotalBsmtSF)] <- medianTotalBsmtSF
AmesData$GarageYrBlt[is.na(AmesData$GarageYrBlt)] <- medianGarageYrBlt
AmesData$GarageArea[is.na(AmesData$GarageArea)] <- medianGarageArea
```

Run a sanity check to make sure the changes took.

```
nan_sums <- colSums(is.na(AmesData))
nan_sums[nan_sums > 0]
```

```
##        Alley     BsmtQual     BsmtCond BsmtExposure BsmtFinType1
##         2727           79           79           79           79
## BsmtFinType2 BsmtFullBath BsmtHalfBath  FireplaceQu   GarageType
##           79            2            2         1422          157
## GarageFinish   GarageCars   GarageQual   GarageCond       PoolQC
##          157            1          158          158         2914
##        Fence  MiscFeature
##         2354         2820
```

```
sum(is.empty(AmesData))
```

```
## [1] 48107
```

There are many other features missing information. A good way to address them is to make a note the feature does not exist in that particular property. For example, properties without basements cannot have values associated with their basements. So we plug those empty values with a string meaning "without".

```
AmesData$Alley[is.na(AmesData$Alley)] <- 'w/o'
AmesData$BsmtQual[is.na(AmesData$BsmtQual)] <- 'w/o'
AmesData$BsmtCond[is.na(AmesData$BsmtCond)] <- 'w/o'
AmesData$BsmtExposure[is.na(AmesData$BsmtExposure)] <- 'w/o'
AmesData$BsmtFinType1[is.na(AmesData$BsmtFinType1)] <- 'w/o'
AmesData$BsmtFinType2[is.na(AmesData$BsmtFinType2)] <- 'w/o'
```

```
AmesData$BsmtFullBath[is.na(AmesData$BsmtFullBath)] <- 'w/o'
AmesData$BsmtHalfBath[is.na(AmesData$BsmtHalfBath)] <- 'w/o'
AmesData$FireplaceQu[is.na(AmesData$FireplaceQu)] <- 'w/o'
AmesData$GarageType[is.na(AmesData$GarageType)] <- 'w/o'
AmesData$GarageFinish[is.na(AmesData$GarageFinish)] <- 'w/o'
AmesData$GarageCars[is.na(AmesData$GarageCars)] <- 'w/o'
AmesData$GarageQual[is.na(AmesData$GarageQual)] <- 'w/o'
AmesData$GarageCond[is.na(AmesData$GarageCond)] <- 'w/o'
AmesData$PoolQC[is.na(AmesData$PoolQC)] <- 'w/o'
AmesData$Fence[is.na(AmesData$Fence)] <- 'w/o'
AmesData$MiscFeature[is.na(AmesData$MiscFeature)] <- 'w/o'
```

The character features need to be converted back to factors for the rest of the analysis. Looking at data structure again confirms a successful conversion.

```
AmesData <- as.data.frame(unclass(AmesData))
str(AmesData)
```

```
## 'data.frame':    2925 obs. of  82 variables:
##  $ Order        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ PID          : int  526301100 526350040 526351010 526353030 527105010 527105030 527127150 5271450
##  $ MSSubClass   : int  20 20 20 20 60 60 120 120 120 60 ...
##  $ MSZoning     : Factor w/ 7 levels "A (agr)","C (all)",..: 6 5 6 6 6 6 6 6 6 6 ...
##  $ LotFrontage  : int  141 80 81 93 74 78 41 43 39 60 ...
##  $ LotArea      : int  31770 11622 14267 11160 13830 9978 4920 5005 5389 7500 ...
##  $ Street       : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Alley        : Factor w/ 3 levels "Grvl","Pave",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ LotShape     : Factor w/ 4 levels "IR1","IR2","IR3",..: 1 4 1 4 1 1 4 1 1 4 ...
##  $ LandContour  : Factor w/ 4 levels "Bnk","HLS","Low",..: 4 4 4 4 4 4 4 2 4 4 ...
##  $ Utilities    : Factor w/ 3 levels "AllPub","NoSeWa",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ LotConfig    : Factor w/ 5 levels "Corner","CulDSac",..: 1 5 1 1 5 5 5 5 5 5 ...
##  $ LandSlope    : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Neighborhood : Factor w/ 28 levels "Blmngtn","Blueste",..: 16 16 16 16 9 9 25 25 25 9 ...
##  $ Condition1   : Factor w/ 9 levels "Artery","Feedr",..: 3 2 3 3 3 3 3 3 3 3 ...
##  $ Condition2   : Factor w/ 8 levels "Artery","Feedr",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ BldgType     : Factor w/ 5 levels "1Fam","2fmCon",..: 1 1 1 1 1 1 5 5 5 1 ...
##  $ HouseStyle   : Factor w/ 8 levels "1.5Fin","1.5Unf",..: 3 3 3 3 6 6 3 3 3 6 ...
##  $ OverallQual  : int  6 5 6 7 5 6 8 8 8 7 ...
##  $ OverallCond  : int  5 6 6 5 5 6 5 5 5 5 ...
##  $ YearBuilt    : int  1960 1961 1958 1968 1997 1998 2001 1992 1995 1999 ...
##  $ YearRemodAdd : int  1960 1961 1958 1968 1998 1998 2001 1992 1996 1999 ...
##  $ RoofStyle    : Factor w/ 6 levels "Flat","Gable",..: 4 2 4 4 2 2 2 2 2 2 ...
##  $ RoofMatl     : Factor w/ 7 levels "CompShg","Membran",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Exterior1st  : Factor w/ 16 levels "AsbShng","AsphShn",..: 4 14 15 4 14 14 6 7 6 14 ...
##  $ Exterior2nd  : Factor w/ 17 levels "AsbShng","AsphShn",..: 11 15 16 4 15 15 6 7 6 15 ...
##  $ MasVnrType   : Factor w/ 6 levels "","BrkCmn","BrkFace",..: 6 5 3 5 5 3 5 5 5 5 ...
##  $ MasVnrArea   : num  112 0 108 0 0 20 0 0 0 0 ...
##  $ ExterQual    : Factor w/ 4 levels "Ex","Fa","Gd",..: 4 4 4 3 4 4 3 3 3 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",..: 2 2 2 2 3 3 3 3 3 3 ...
##  $ BsmtQual     : Factor w/ 7 levels "","Ex","Fa","Gd",..: 6 6 6 6 4 6 4 4 4 6 ...
##  $ BsmtCond     : Factor w/ 7 levels "","Ex","Fa","Gd",..: 4 6 6 6 6 6 6 6 6 6 ...
##  $ BsmtExposure : Factor w/ 6 levels "","Av","Gd","Mn",..: 3 5 5 5 5 5 4 5 5 5 ...
##  $ BsmtFinType1 : Factor w/ 8 levels "","ALQ","BLQ",..: 3 6 2 2 4 4 4 2 4 7 ...
##  $ BsmtFinSF1   : num  639 468 923 1065 791 ...
```

```
##  $ BsmtFinType2 : Factor w/ 8 levels "","ALQ","BLQ",..: 7 5 7 7 7 7 7 7 7 7 ...
##  $ BsmtFinSF2   : num  0 144 0 0 0 0 0 0 0 0 ...
##  $ BsmtUnfSF    : num  441 270 406 1045 137 ...
##  $ TotalBsmtSF  : num  1080 882 1329 2110 928 ...
##  $ 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",..: 2 5 5 1 3 1 1 1 1 3 ...
##  $ CentralAir   : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Electrical   : Factor w/ 6 levels "","FuseA","FuseF",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ X1stFlrSF    : int  1656 896 1329 2110 928 926 1338 1280 1616 1028 ...
##  $ X2ndFlrSF    : int  0 0 0 0 701 678 0 0 0 776 ...
##  $ LowQualFinSF : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea    : int  1656 896 1329 2110 1629 1604 1338 1280 1616 1804 ...
##  $ BsmtFullBath : Factor w/ 5 levels "0","1","2","3",..: 2 1 1 2 1 1 2 1 2 1 ...
##  $ BsmtHalfBath : Factor w/ 4 levels "0","1","2","w/o": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FullBath     : int  1 1 1 2 2 2 2 2 2 2 ...
##  $ HalfBath     : int  0 0 1 1 1 1 0 0 0 1 ...
##  $ BedroomAbvGr : int  3 2 3 3 3 3 2 2 2 3 ...
##  $ KitchenAbvGr : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ KitchenQual  : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 3 1 5 3 3 3 3 3 ...
##  $ TotRmsAbvGrd : int  7 5 6 8 6 7 6 5 5 7 ...
##  $ Functional   : Factor w/ 8 levels "Maj1","Maj2",..: 8 8 8 8 8 8 8 8 8 8 ...
##  $ Fireplaces   : int  2 0 0 2 1 1 0 0 1 1 ...
##  $ FireplaceQu  : Factor w/ 6 levels "Ex","Fa","Gd",..: 3 6 6 5 5 3 6 6 5 5 ...
##  $ GarageType   : Factor w/ 7 levels "2Types","Attchd",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ GarageYrBlt  : num  1960 1961 1958 1968 1997 ...
##  $ GarageFinish : Factor w/ 5 levels "","Fin","RFn",..: 2 4 4 2 2 2 2 3 3 2 ...
##  $ GarageCars   : Factor w/ 7 levels "0","1","2","3",..: 3 2 2 3 3 3 3 3 3 3 ...
##  $ GarageArea   : num  528 730 312 522 482 470 582 506 608 442 ...
##  $ GarageQual   : Factor w/ 7 levels "","Ex","Fa","Gd",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ GarageCond   : Factor w/ 7 levels "","Ex","Fa","Gd",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ PavedDrive   : Factor w/ 3 levels "N","P","Y": 2 3 3 3 3 3 3 3 3 3 ...
##  $ WoodDeckSF   : int  210 140 393 0 212 360 0 0 237 140 ...
##  $ OpenPorchSF  : int  62 0 36 0 34 36 0 82 152 60 ...
##  $ EnclosedPorch: int  0 0 0 0 0 0 170 0 0 0 ...
##  $ X3SsnPorch   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ScreenPorch  : int  0 120 0 0 0 0 0 144 0 0 ...
##  $ PoolArea     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC       : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ Fence        : Factor w/ 5 levels "GdPrv","GdWo",..: 5 3 5 5 3 5 5 5 5 5 ...
##  $ MiscFeature  : Factor w/ 5 levels "Gar2","Othr",..: 5 5 1 5 5 5 5 5 5 5 ...
##  $ MiscVal      : int  0 0 12500 0 0 0 0 0 0 0 ...
##  $ MoSold       : int  5 6 6 4 3 6 4 1 3 6 ...
##  $ YrSold       : int  2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
##  $ SaleType     : Factor w/ 10 levels "COD","Con","ConLD",..: 10 10 10 10 10 10 10 10 10 10 ...
##  $ SaleCondition: Factor w/ 6 levels "Abnorml","AdjLand",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ SalePrice    : int  215000 105000 172000 244000 189900 195500 213500 191500 236500 189000 ...
```

The goal isn't to remove all empty values. Many are legitimate data that should be preserved. Run a count to see how many remain.

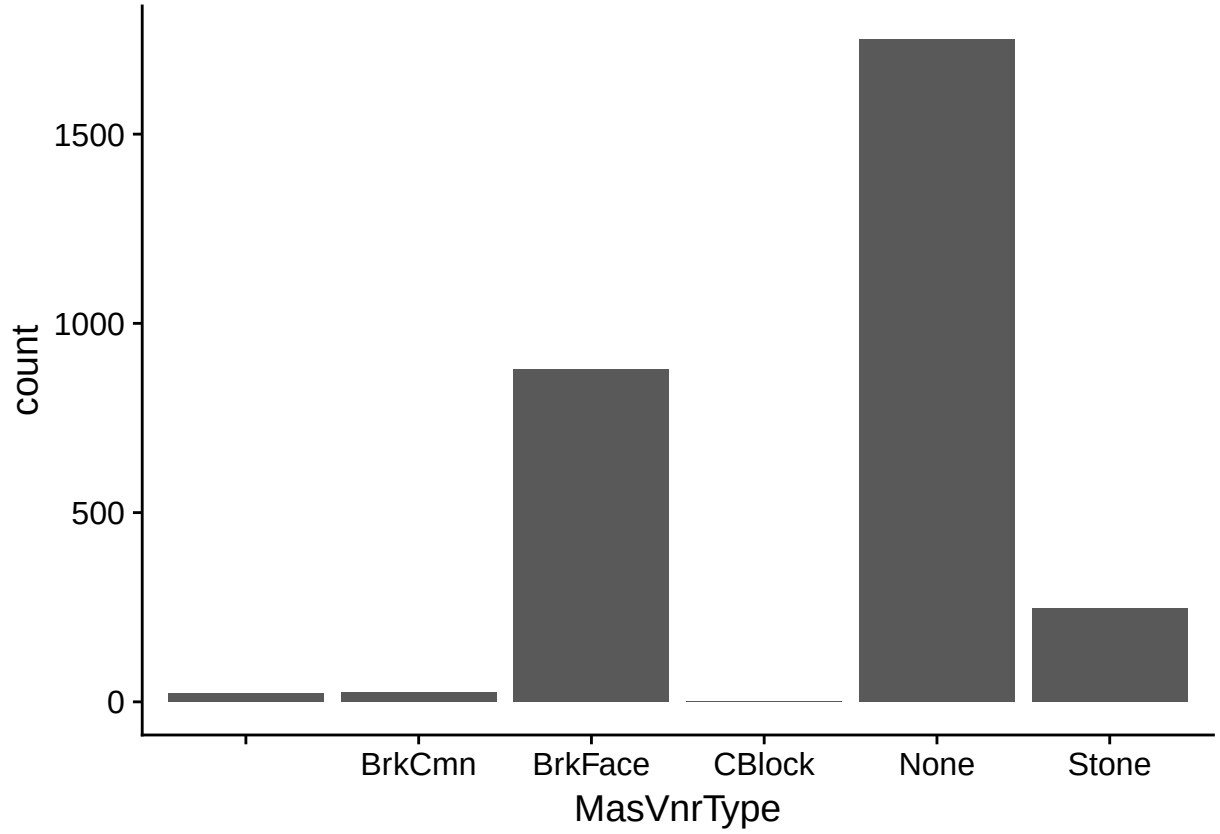
```
sum(is.empty(AmesData))
```

```
## [1] 30190
```

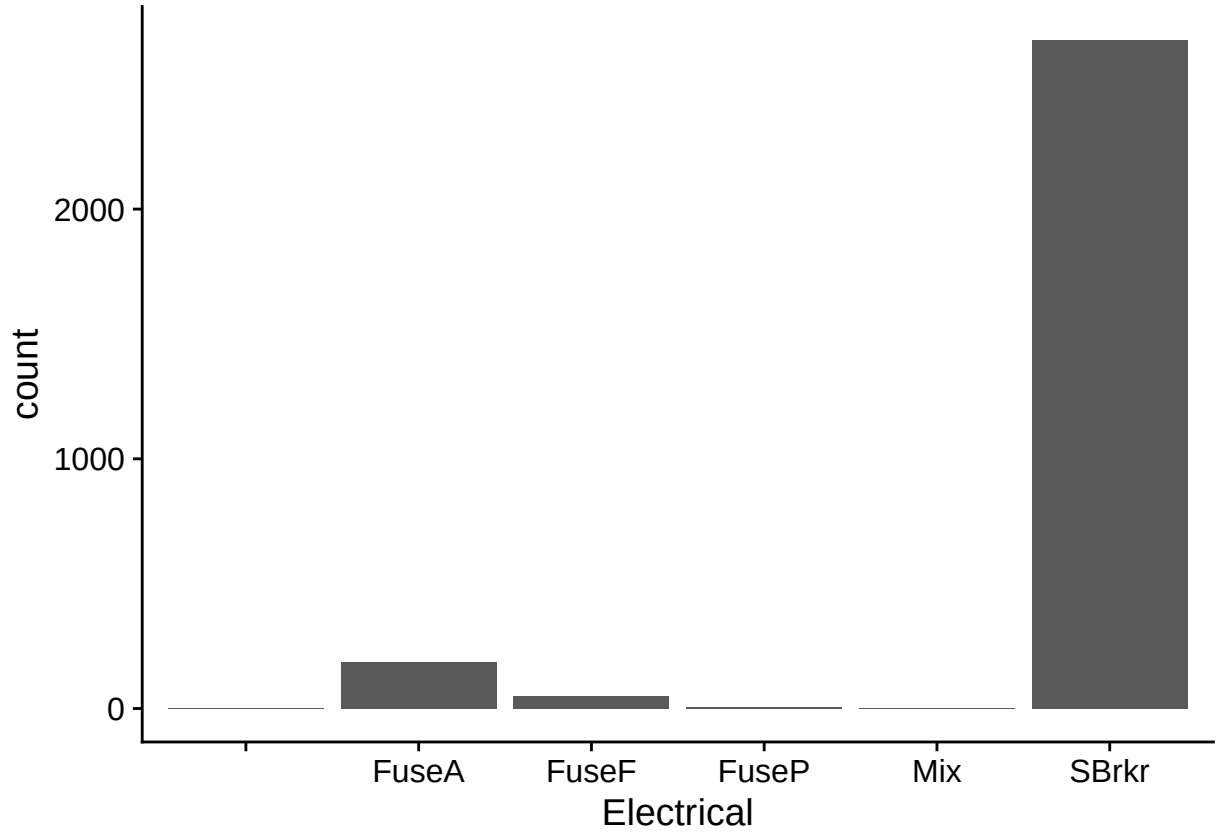
There are two features that contain empty values that cannot be treated as "withouts" or with median values. For those – *MasVnrType* and *Electrical* – we need to find alternative values. We cannot say a property is

without an electrical system but we can use the most common type of electrical system as a proxy.

```
ggplot(AmesData, aes(x = MasVnrType)) + geom_histogram(stat = 'count')
```




```
ggplot(AmesData, aes(x = Electrical)) + geom_histogram(stat = 'count')
```



It looks like the most common type of *MasVnrType* is "None." For *Electrical* it looks like "SBrkr" so we can use those to replace the empty values. We run a count to make sure the changes took hold.

```
AmesData$MasVnrType[AmesData$MasVnrType == ''] <- 'None'
AmesData$Electrical[AmesData$Electrical == ''] <- 'SBrkr'
nan_sums <- colSums(is.na(AmesData))
nan_sums[nan_sums > 0]
```

```
## named numeric(0)
```

```
sum(is.empty(AmesData$Electrical))
```

```
## [1] 0
```

```
sum(is.empty(AmesData$MasVnrType))
```

```
## [1] 0
```

```
sum(is.empty(AmesData))
```

```
## [1] 30190
```

Data Explanations

For the remaining operations, particular the graphics involving the *ggplot2* package, it will be helpful to convert the data into the *tibble* data frame format for ease of use.

```
AmesTibble <- as_tibble(AmesData)
AmesTibble
```

```
## # A tibble: 2,925 x 82
##    Order       PID MSSubClass MSZoning LotFrontage LotArea Street Alley
##    <int>     <int>      <int> <fct>          <int>   <int> <fct>  <fct>
##  1     1 526301100         20 RL               141   31770 Pave   w/o
##  2     2 526350040         20 RH                80   11622 Pave   w/o
##  3     3 526351010         20 RL                81   14267 Pave   w/o
##  4     4 526353030         20 RL                93   11160 Pave   w/o
##  5     5 527105010         60 RL                74   13830 Pave   w/o
##  6     6 527105030         60 RL                78    9978 Pave   w/o
##  7     7 527127150        120 RL                41    4920 Pave   w/o
##  8     8 527145080        120 RL                43    5005 Pave   w/o
##  9     9 527146030        120 RL                39    5389 Pave   w/o
## 10    10 527162130         60 RL                60    7500 Pave   w/o
## # ... with 2,915 more rows, and 74 more variables: LotShape <fct>,
## #   LandContour <fct>, Utilities <fct>, LotConfig <fct>, LandSlope <fct>,
## #   Neighborhood <fct>, Condition1 <fct>, Condition2 <fct>,
## #   BldgType <fct>, HouseStyle <fct>, OverallQual <int>,
## #   OverallCond <int>, YearBuilt <int>, YearRemodAdd <int>,
## #   RoofStyle <fct>, RoofMatl <fct>, Exterior1st <fct>, Exterior2nd <fct>,
## #   MasVnrType <fct>, MasVnrArea <dbl>, ExterQual <fct>, ExterCond <fct>,
## #   Foundation <fct>, BsmtQual <fct>, BsmtCond <fct>, BsmtExposure <fct>,
## #   BsmtFinType1 <fct>, BsmtFinSF1 <dbl>, BsmtFinType2 <fct>,
## #   BsmtFinSF2 <dbl>, BsmtUnfSF <dbl>, TotalBsmtSF <dbl>, Heating <fct>,
## #   HeatingQC <fct>, CentralAir <fct>, Electrical <fct>, X1stFlrSF <int>,
## #   X2ndFlrSF <int>, LowQualFinSF <int>, GrLivArea <int>,
## #   BsmtFullBath <fct>, BsmtHalfBath <fct>, FullBath <int>,
## #   HalfBath <int>, BedroomAbvGr <int>, KitchenAbvGr <int>,
## #   KitchenQual <fct>, TotRmsAbvGrd <int>, Functional <fct>,
## #   Fireplaces <int>, FireplaceQu <fct>, GarageType <fct>,
## #   GarageYrBlt <dbl>, GarageFinish <fct>, GarageCars <fct>,
## #   GarageArea <dbl>, GarageQual <fct>, GarageCond <fct>,
## #   PavedDrive <fct>, WoodDeckSF <int>, OpenPorchSF <int>,
## #   EnclosedPorch <int>, X3SsnPorch <int>, ScreenPorch <int>,
## #   PoolArea <int>, PoolQC <fct>, Fence <fct>, MiscFeature <fct>,
## #   MiscVal <int>, MoSold <int>, YrSold <int>, SaleType <fct>,
## #   SaleCondition <fct>, SalePrice <int>
```

Contingency tables are useful to get a sense of how many levels exist for different features, and how many properties fall into each category. Contingency tables can count information or percentages of total instances. Graphic representations of the data in the tables are also useful. Contingecy tables and histograms help us understand the frequency of features, or how often we encounter them in the data set.

In this way, we can see if any particular housing features dominate in the Ames real estate market. We get an understanding of what a typical or representative property might look like.

To illustrate, here are two contingency tables for the *OverallCond* feature, along with a histogram of the distribution.

```
table(AmesTibble$OverallCond)
```

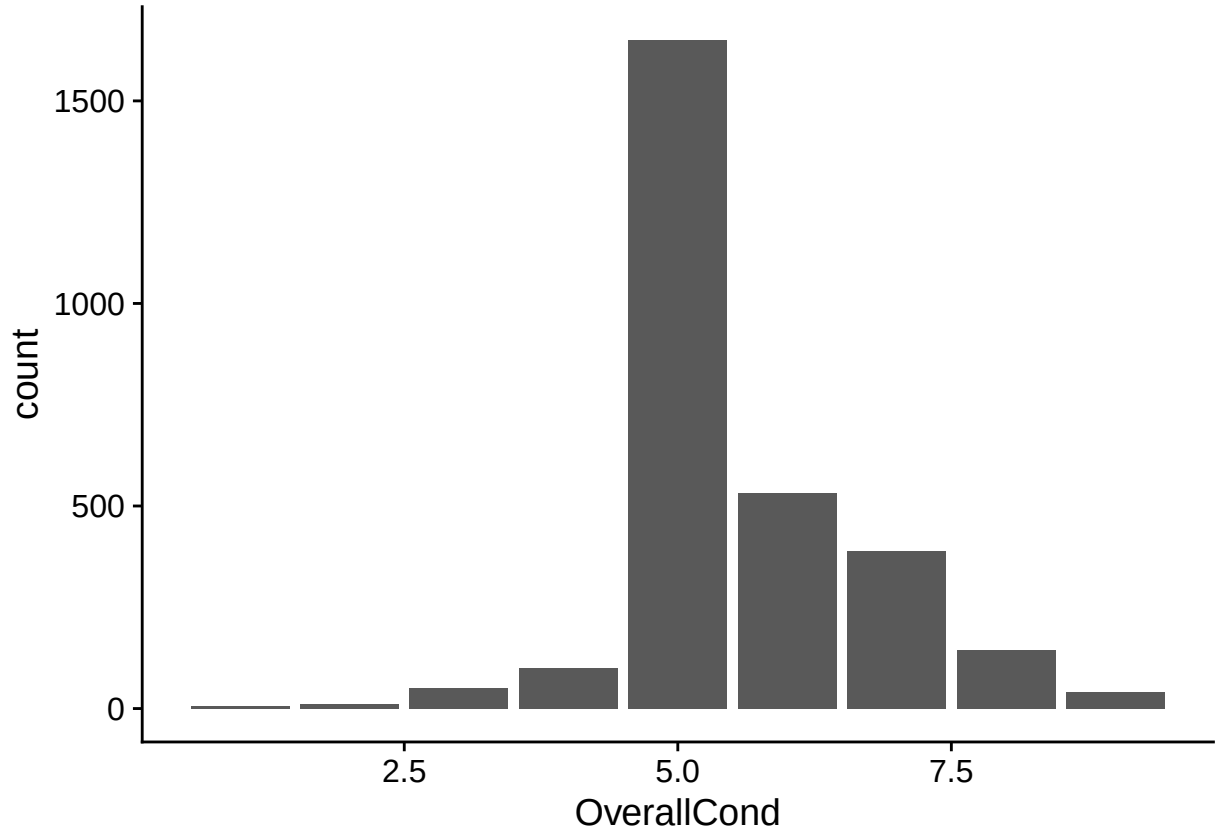
```
##
##    1    2    3    4    5    6    7    8    9
##    7   10   50  101 1650  532  390  144   41
```

```
round(table(AmesTibble$OverallCond)/length(AmesTibble$OverallCond), 3)
```

```
##
```

```
##     1     2     3     4     5     6     7     8     9
## 0.002 0.003 0.017 0.035 0.564 0.182 0.133 0.049 0.014
```

```
ggplot(AmesData, aes(x = OverallCond)) + geom_histogram(stat = 'count')
```



The table on *OverallCond* shows what we might expect – most houses are in the middle. They are in average condition.

Roof features – the style of roof and the roofing material used – are also of interest. They tell us something about the architecture and construction of properties. Here are contingency tables and histrograms of each.

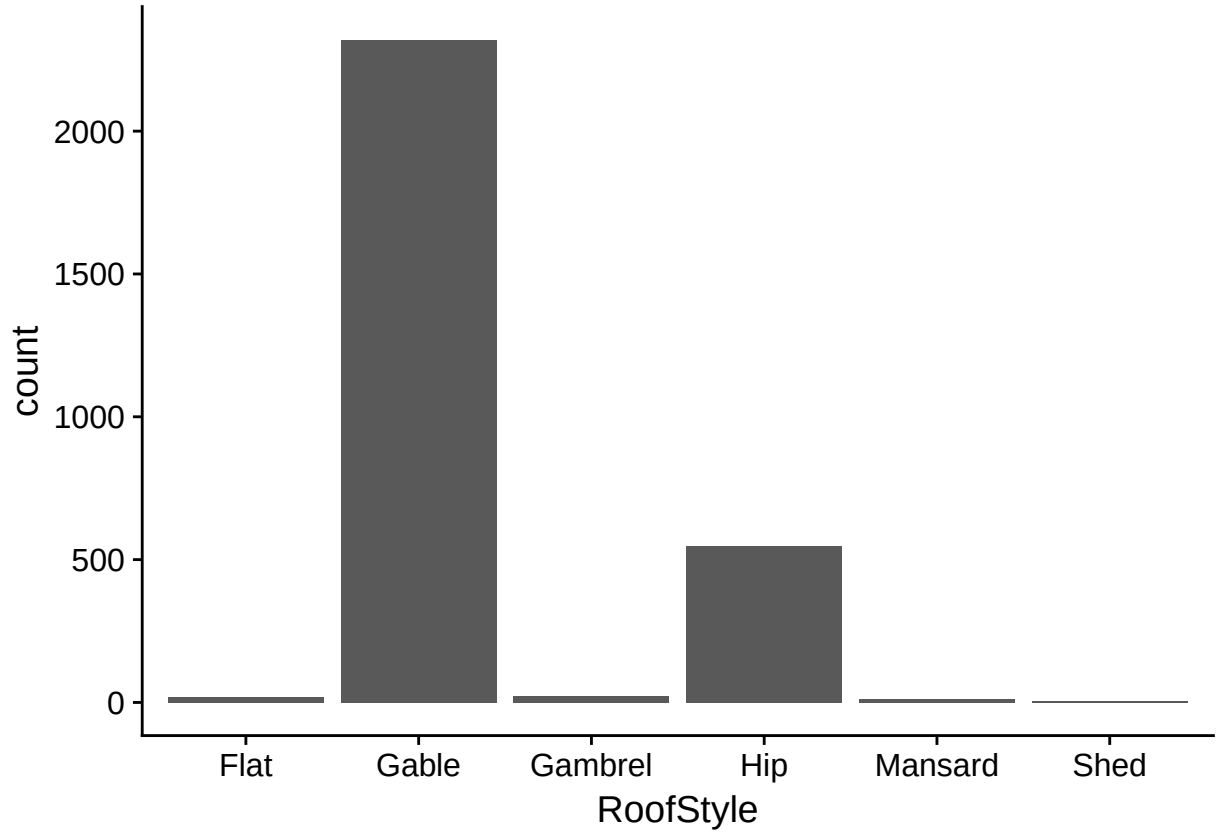
```
table(AmesTibble$RoofStyle)
```

```
##
##    Flat   Gable Gambrel     Hip Mansard    Shed
##      20    2320      22     547      11       5
```

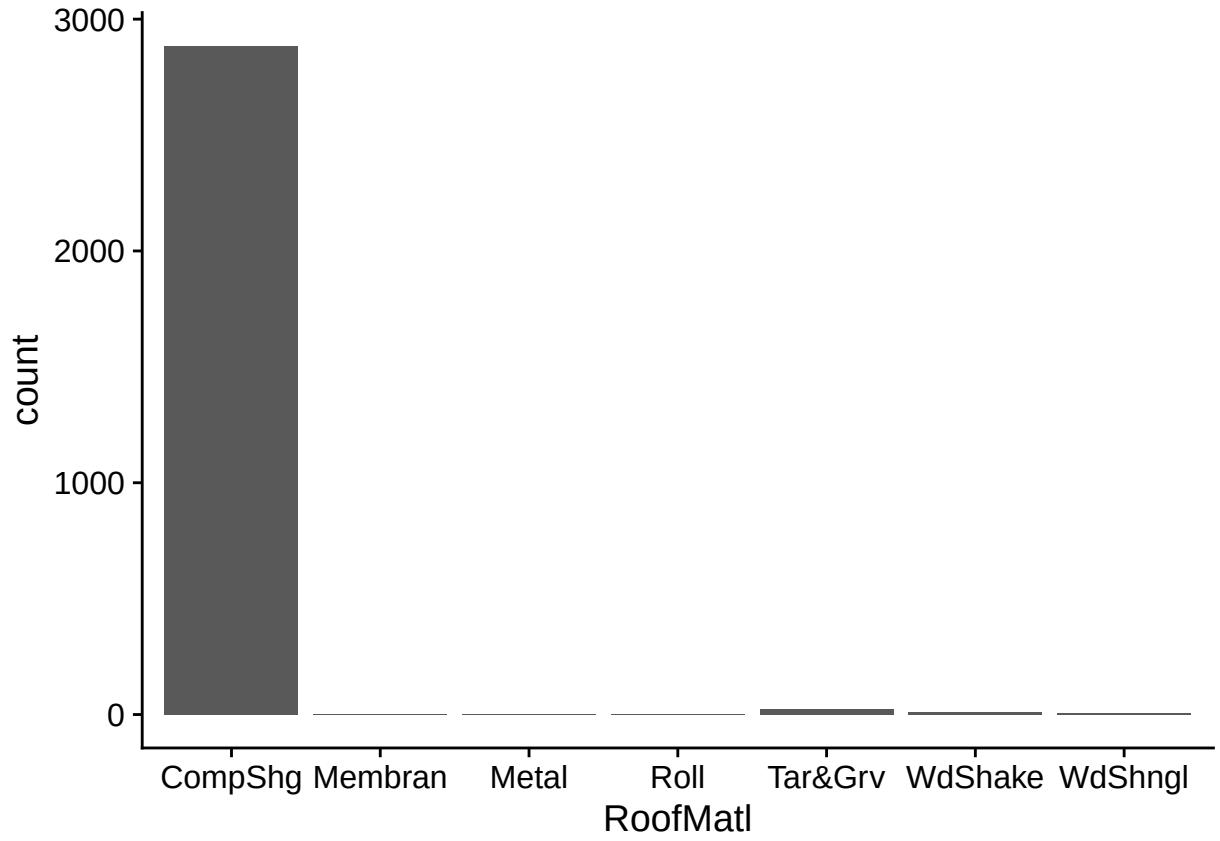
```
round(table(AmesTibble$RoofStyle)/length(AmesTibble$RoofStyle), 3)
```

```
##
##    Flat   Gable Gambrel     Hip Mansard    Shed
##   0.007   0.793   0.008   0.187   0.004   0.002
```

```
ggplot(AmesTibble, aes(x = RoofStyle)) + geom_histogram(stat = 'count')
```




```
ggplot(AmesTibble, aes(x = RoofMatl)) + geom_histogram(stat = 'count')
```



What we see is two types of structure – Gable and Hip – are prevalent in architectural style of Ames houses. Gable dominates by far. We also see that virtually all rooves are made of composite shingles.

Since roof features can be broken down into two subcategories – style and material – we can classify rooves both ways and construct a table to represent what materials are associated with which styles.

```
RoofTable <- table(AmesTibble$RoofStyle, AmesTibble$RoofMatl)
DecTable <- table(AmesTibble$RoofStyle, AmesTibble$RoofMatl)/length(AmesData$RoofStyle)
DecTable <- round(DecTable, 3)
RoofTable
```

```
##
##           CompShg Membran Metal Roll Tar&Grv WdShake WdShngl
##   Flat          1       1     1    0      17       0       0
##   Gable      2309       0     0    1       6       0       4
##   Gambrel      22       0     0    0       0       0       0
##   Hip         541       0     0    0       0       5       1
##   Mansard       8       0     0    0       0       3       0
##   Shed          3       0     0    0       0       1       1
```

```
DecTable
```

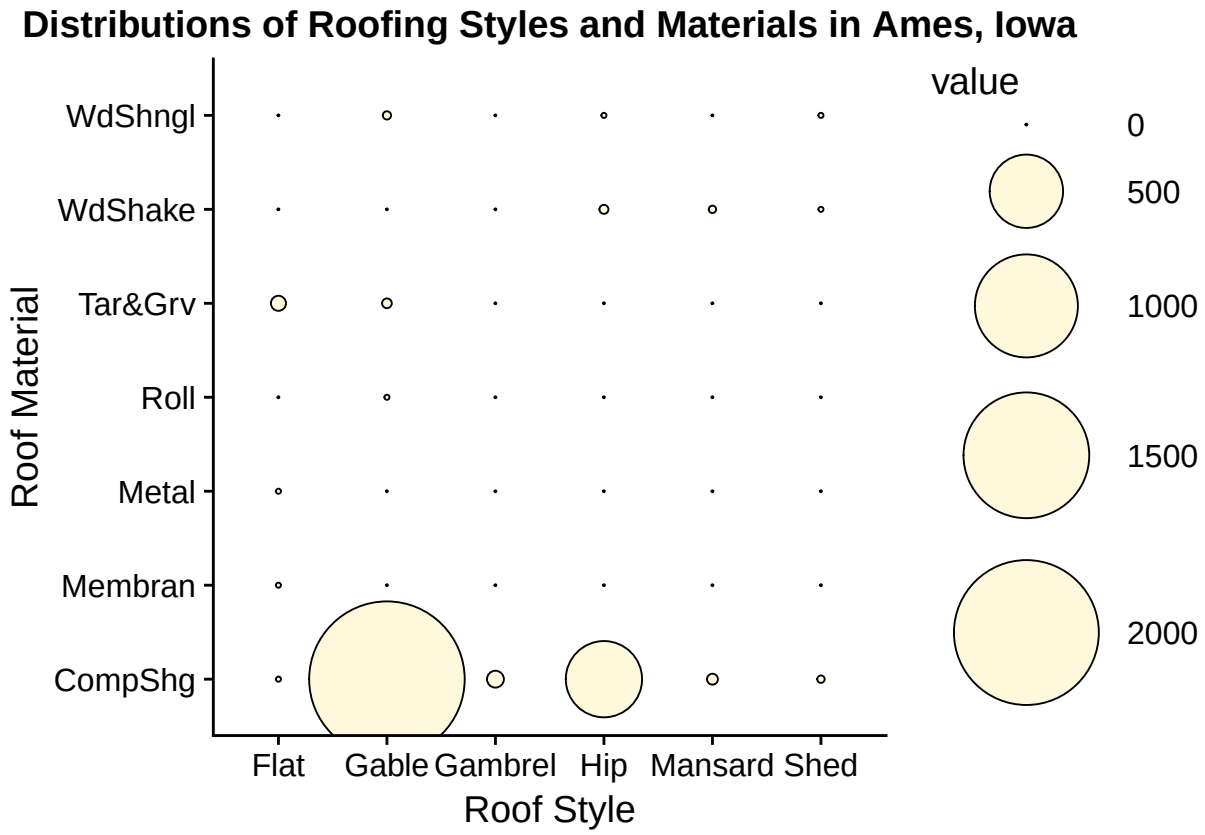
```
##
##           CompShg Membran Metal  Roll Tar&Grv WdShake WdShngl
##   Flat      0.000   0.000 0.000 0.000   0.006   0.000   0.000
##   Gable     0.789   0.000 0.000 0.000   0.002   0.000   0.001
##   Gambrel   0.008   0.000 0.000 0.000   0.000   0.000   0.000
##   Hip       0.185   0.000 0.000 0.000   0.000   0.002   0.000
##   Mansard   0.003   0.000 0.000 0.000   0.000   0.001   0.000
```

```
##    Shed       0.001    0.000 0.000 0.000    0.000    0.000    0.000
```

The tables above give us the count and proportions of each combination of roofing style and material in the Ames market. It seems the choices are pretty limited there.

A graphic way to represent this information is a balloon plot. The size of the balloon indicates the count associated with each combination of style and material.

```
AmesRoofMelt <- melt(RoofTable)
ggplot(AmesRoofMelt, aes(Var.1, Var.2)) +
        geom_point(aes(size = value), shape = 21, color = 'black', fill = 'cornsilk') +
        scale_size_area(max_size = 27) +
        ggtitle('Distributions of Roofing Styles and Materials in Ames, Iowa') +
        labs(x = 'Roof Style', y = 'Roof Material') +
        theme(panel.background = element_blank(),
              panel.border = element_rect(color = 'blue', fill = NA, size = 1))
```

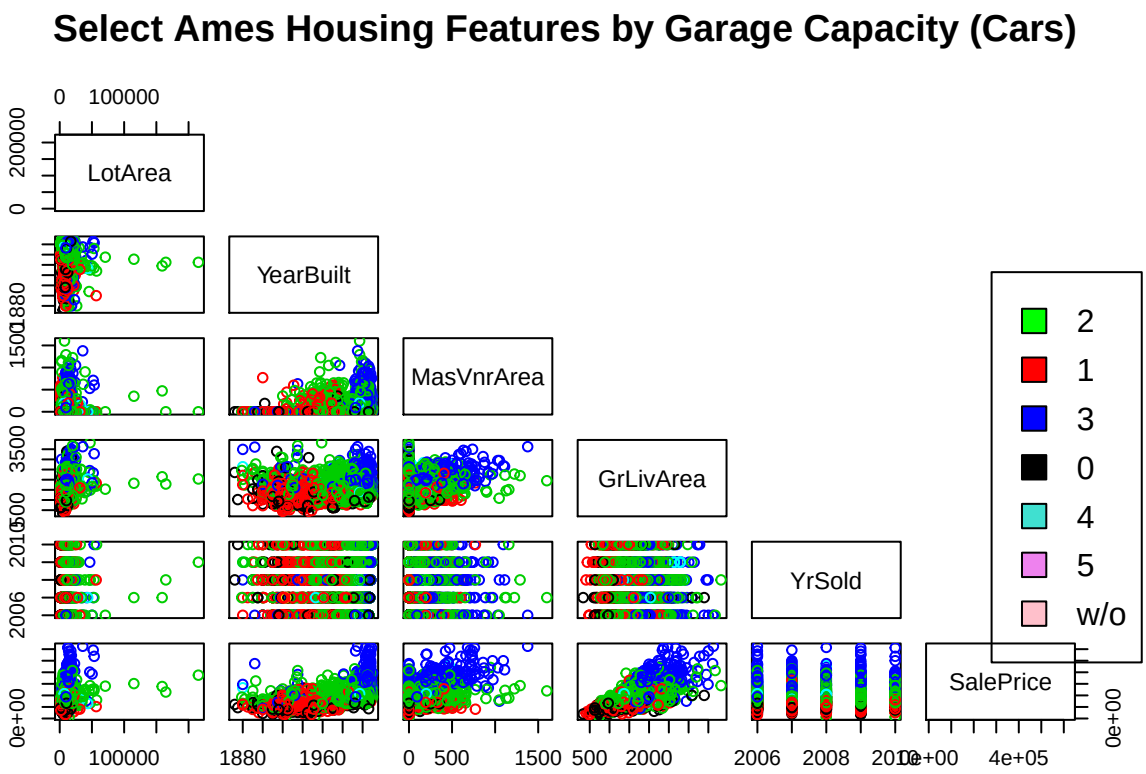


The balloon plot is a visual representation of the same information contained in the tables preceding it.

We can compare any two features against each other to get a sense of the relationship between them. This is not practical for all features in such a large data set, and many comparisons would be meaningless, but we can select some features of interest and look for patterns there. The pair plots below consider six numerical features. Colors indiate the size of the garages associated with each property, in terms of the number of cars each garage can hold.

For our purpose, the *SalePrice* is the target for our analysis. We want to know how the other variables impact the value of the home when it comes time to sell.

```
AmesPairs <- AmesTibble[c(6, 21, 28, 48, 79, 82)]
pairs(AmesPairs, upper.panel = NULL, col = AmesTibble$GarageCars,
      main = 'Select Ames Housing Features by Garage Capacity (Cars)')
par(xpd = TRUE)
legend(0.9, 0.7, as.vector(unique(AmesTibble$GarageCars)),
       fill = c('green', 'red', 'blue', 'black', 'turquoise', 'violet', 'pink'))
```

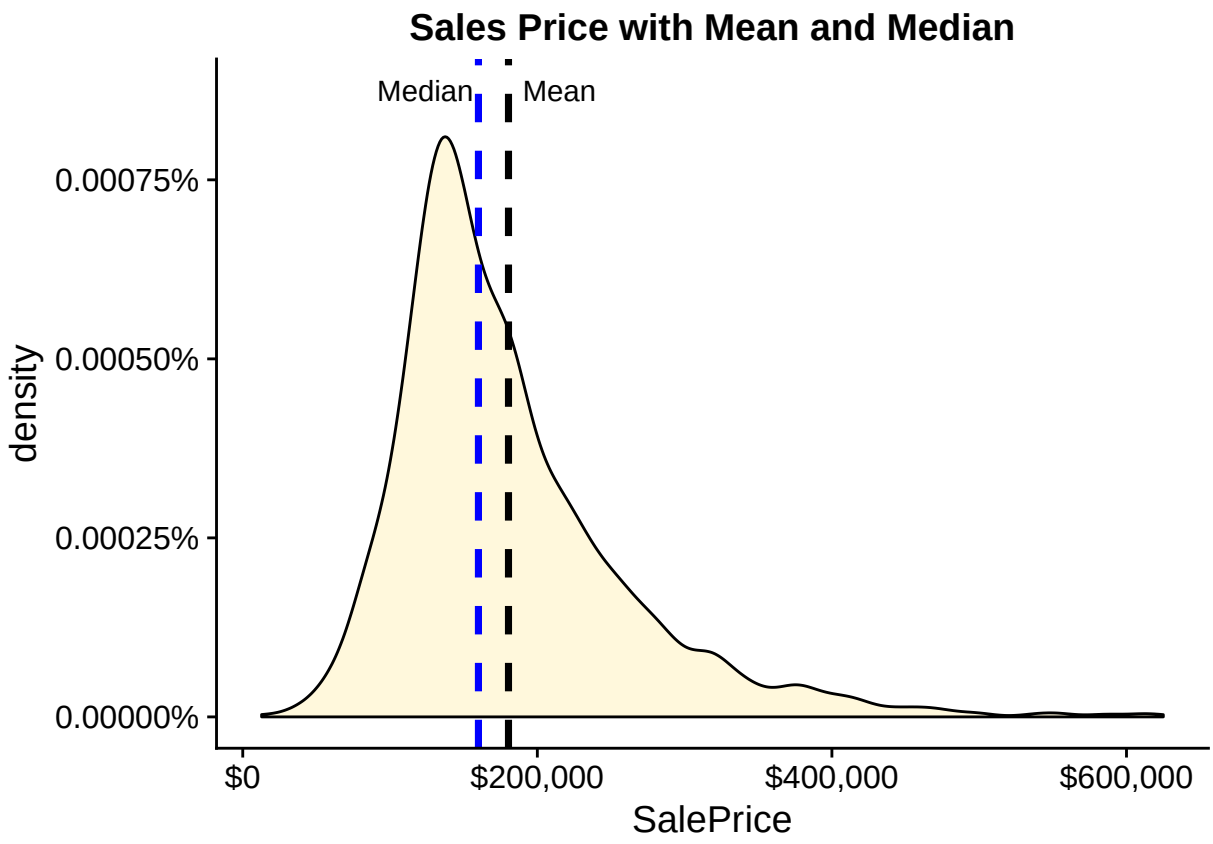


The pair plots show some general linear relationships exist between these variables. It appears *LotArea* has some outliers which we might consider removing since they tend to dominate the scale. It also seems the tightest linear relationship exists between *GrLivArea* and *SalePrice*. All the relationships seem to be positive, that is, as the x-value increases, so does *SalePrice*. It also appears that the more expensive homes have larger garages, as do homes with larger living areas, and those built more recently.

The *SalePrice* is the target feature in the *Ames* data set. It is possible to get a sense of the distribution of sales prices between 2006 and 2010 for the entire area. The following graph accomplishes this by aggregating all the sales data. Though the distribution looks slightly normalish, there is definitely left skew, since the mean exceeds the median. In addition, the mode does not equal the mean and median. This is not a normal distribution.

```
ggplot(AmesTibble, aes(x = SalePrice)) + geom_density(fill = 'cornsilk') +
        geom_vline(aes(xintercept = mean(SalePrice)), color = 'black',
                   linetype = 'dashed', size = 1.25) +
        geom_vline(aes(xintercept = median(SalePrice)), color = 'blue',
                   linetype = 'dashed', size = 1.25) +
        ggtitle('Sales Price with Mean and Median') +
        annotate(geom = 'text', x = 215000, y = 0.00000875, label = 'Mean') +
        annotate(geom = 'text', x = 124000, y = 0.00000875, label = 'Median') +
```

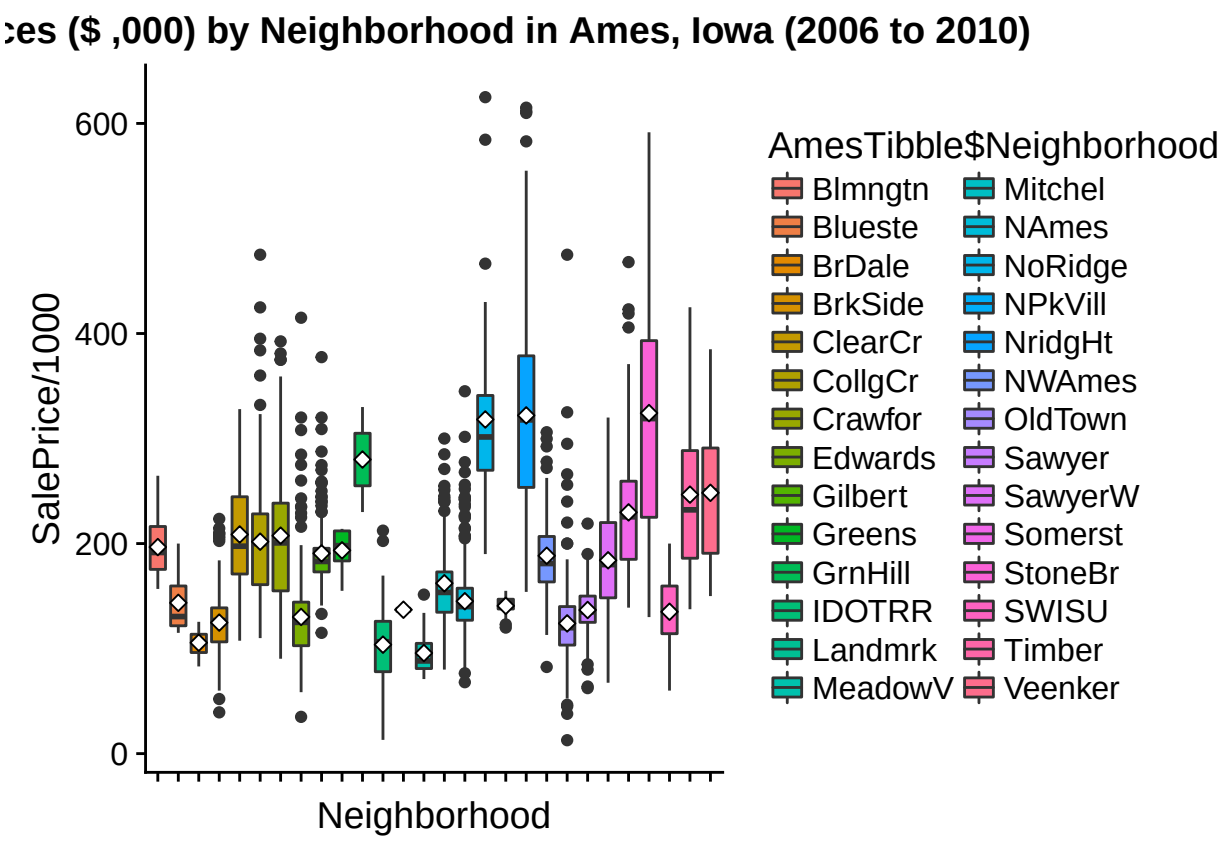
```
        scale_x_continuous(labels = scales::dollar) +
        scale_y_continuous(labels = scales::percent)
```



Most homes in Ames seem to sell in the range of $150,000 to $200,000 though many homes are evidently worth twice or three to four times that range.

The previous graph gives an idea of the aggregate distribution of *SalePrice* for all properties in the city of Ames. Another way to look at the distribution of sales prices is by neighborhood. Box plots are useful for this. The white diamond in the body of the box plot indicates mean value. *SalePrice* is numeric while *Neighborhood* is a factor.

```
ggplot(AmesTibble, aes(x = Neighborhood, y = SalePrice/1000)) +
        geom_boxplot(aes(fill = AmesTibble$Neighborhood)) +
        stat_summary(fun.y = 'mean', geom = 'point', shape = 23, size = 2,
                     fill = 'white') +
        ggtitle("Sales Prices ($ ,000) by Neighborhood in Ames, Iowa (2006 to 2010)") +
        theme(axis.text.x=element_blank())
```



We can see in the box plots which neighborhoods are the least and most popular, the least and most valuable, and which contain mostly less expensive homes, or the more expensive ones. This detail is not available in the density chart above, which give an average value for all homes. Here, we see the average value and the range of values by neighborhood.

Determining the Importance of Features

Many factors in the data need to be converted to dummy variables in order to include them in the analysis. Dummy variables are used to allow non-numeric features to be used in statistical analyses. Look at the data structure afterwards to make sure the changes occurred and the new dummy columns were added.

```
AmesDummy <- dummy_cols(AmesTibble, remove_first_dummy = TRUE)
str(AmesDummy)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    2925 obs. of  340 variables:
##  $ Order                : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ PID                  : int  526301100 526350040 526351010 526353030 527105010 527105030 527127150
##  $ MSSubClass           : int  20 20 20 20 60 60 120 120 120 60 ...
##  $ MSZoning             : Factor w/ 7 levels "A (agr)","C (all)",..: 6 5 6 6 6 6 6 6 6 6 ...
##  $ LotFrontage          : int  141 80 81 93 74 78 41 43 39 60 ...
##  $ LotArea              : int  31770 11622 14267 11160 13830 9978 4920 5005 5389 7500 ...
##  $ Street               : Factor w/ 2 levels "Grvl","Pave": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Alley                : Factor w/ 3 levels "Grvl","Pave",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ LotShape             : Factor w/ 4 levels "IR1","IR2","IR3",..: 1 4 1 4 1 1 4 1 1 4 ...
##  $ LandContour          : Factor w/ 4 levels "Bnk","HLS","Low",..: 4 4 4 4 4 4 4 2 4 4 ...
##  $ Utilities            : Factor w/ 3 levels "AllPub","NoSeWa",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
##  $ LotConfig           : Factor w/ 5 levels "Corner","CulDSac",..: 1 5 1 1 5 5 5 5 5 5 ...
##  $ LandSlope           : Factor w/ 3 levels "Gtl","Mod","Sev": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Neighborhood        : Factor w/ 28 levels "Blmngtn","Blueste",..: 16 16 16 16 9 9 25 25 25 9 ...
##  $ Condition1          : Factor w/ 9 levels "Artery","Feedr",..: 3 2 3 3 3 3 3 3 3 3 ...
##  $ Condition2          : Factor w/ 8 levels "Artery","Feedr",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ BldgType            : Factor w/ 5 levels "1Fam","2fmCon",..: 1 1 1 1 1 1 5 5 5 1 ...
##  $ HouseStyle          : Factor w/ 8 levels "1.5Fin","1.5Unf",..: 3 3 3 3 6 6 3 3 3 6 ...
##  $ OverallQual         : int  6 5 6 7 5 6 8 8 8 7 ...
##  $ OverallCond         : int  5 6 6 5 5 6 5 5 5 5 ...
##  $ YearBuilt           : int  1960 1961 1958 1968 1997 1998 2001 1992 1995 1999 ...
##  $ YearRemodAdd        : int  1960 1961 1958 1968 1998 1998 2001 1992 1996 1999 ...
##  $ RoofStyle           : Factor w/ 6 levels "Flat","Gable",..: 4 2 4 4 2 2 2 2 2 2 ...
##  $ RoofMatl            : Factor w/ 7 levels "CompShg","Membran",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Exterior1st         : Factor w/ 16 levels "AsbShng","AsphShn",..: 4 14 15 4 14 14 6 7 6 14 ...
##  $ Exterior2nd         : Factor w/ 17 levels "AsbShng","AsphShn",..: 11 15 16 4 15 15 6 7 6 15 ...
##  $ MasVnrType          : Factor w/ 6 levels "","BrkCmn","BrkFace",..: 6 5 3 5 5 3 5 5 5 5 ...
##  $ MasVnrArea          : num  112 0 108 0 0 20 0 0 0 0 ...
##  $ ExterQual           : Factor w/ 4 levels "Ex","Fa","Gd",..: 4 4 4 3 4 4 3 3 3 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",..: 2 2 2 2 3 3 3 3 3 3 ...
##  $ BsmtQual            : Factor w/ 7 levels "","Ex","Fa","Gd",..: 6 6 6 6 4 6 4 4 4 6 ...
##  $ BsmtCond            : Factor w/ 7 levels "","Ex","Fa","Gd",..: 4 6 6 6 6 6 6 6 6 6 ...
##  $ BsmtExposure        : Factor w/ 6 levels "","Av","Gd","Mn",..: 3 5 5 5 5 5 4 5 5 5 ...
##  $ BsmtFinType1        : Factor w/ 8 levels "","ALQ","BLQ",..: 3 6 2 2 4 4 4 2 4 7 ...
##  $ BsmtFinSF1          : num  639 468 923 1065 791 ...
##  $ BsmtFinType2        : Factor w/ 8 levels "","ALQ","BLQ",..: 7 5 7 7 7 7 7 7 7 7 ...
##  $ BsmtFinSF2          : num  0 144 0 0 0 0 0 0 0 0 ...
##  $ BsmtUnfSF           : num  441 270 406 1045 137 ...
##  $ TotalBsmtSF         : num  1080 882 1329 2110 928 ...
##  $ 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",..: 2 5 5 1 3 1 1 1 1 3 ...
##  $ CentralAir          : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Electrical          : Factor w/ 6 levels "","FuseA","FuseF",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ X1stFlrSF           : int  1656 896 1329 2110 928 926 1338 1280 1616 1028 ...
##  $ X2ndFlrSF           : int  0 0 0 0 701 678 0 0 0 776 ...
##  $ LowQualFinSF        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GrLivArea           : int  1656 896 1329 2110 1629 1604 1338 1280 1616 1804 ...
##  $ BsmtFullBath        : Factor w/ 5 levels "0","1","2","3",..: 2 1 1 2 1 1 2 1 2 1 ...
##  $ BsmtHalfBath        : Factor w/ 4 levels "0","1","2","w/o": 1 1 1 1 1 1 1 1 1 1 ...
##  $ FullBath            : int  1 1 1 2 2 2 2 2 2 2 ...
##  $ HalfBath            : int  0 0 1 1 1 1 0 0 0 1 ...
##  $ BedroomAbvGr        : int  3 2 3 3 3 3 2 2 2 3 ...
##  $ KitchenAbvGr        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ KitchenQual         : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 3 1 5 3 3 3 3 3 ...
##  $ TotRmsAbvGrd        : int  7 5 6 8 6 7 6 5 5 7 ...
##  $ Functional          : Factor w/ 8 levels "Maj1","Maj2",..: 8 8 8 8 8 8 8 8 8 8 ...
##  $ Fireplaces          : int  2 0 0 2 1 1 0 0 1 1 ...
##  $ FireplaceQu         : Factor w/ 6 levels "Ex","Fa","Gd",..: 3 6 6 5 5 3 6 6 5 5 ...
##  $ GarageType          : Factor w/ 7 levels "2Types","Attchd",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ GarageYrBlt         : num  1960 1961 1958 1968 1997 ...
##  $ GarageFinish        : Factor w/ 5 levels "","Fin","RFn",..: 2 4 4 2 2 2 2 3 3 2 ...
##  $ GarageCars          : Factor w/ 7 levels "0","1","2","3",..: 3 2 2 3 3 3 3 3 3 3 ...
##  $ GarageArea          : num  528 730 312 522 482 470 582 506 608 442 ...
##  $ GarageQual          : Factor w/ 7 levels "","Ex","Fa","Gd",..: 6 6 6 6 6 6 6 6 6 6 ...
```

```
##  $ GarageCond           : Factor w/ 7 levels "","Ex","Fa","Gd",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ PavedDrive           : Factor w/ 3 levels "N","P","Y": 2 3 3 3 3 3 3 3 3 3 ...
##  $ WoodDeckSF           : int  210 140 393 0 212 360 0 0 237 140 ...
##  $ OpenPorchSF          : int  62 0 36 0 34 36 0 82 152 60 ...
##  $ EnclosedPorch        : int  0 0 0 0 0 0 170 0 0 0 ...
##  $ X3SsnPorch           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ScreenPorch          : int  0 120 0 0 0 0 0 144 0 0 ...
##  $ PoolArea             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PoolQC               : Factor w/ 5 levels "Ex","Fa","Gd",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ Fence                : Factor w/ 5 levels "GdPrv","GdWo",..: 5 3 5 5 3 5 5 5 5 5 ...
##  $ MiscFeature          : Factor w/ 5 levels "Gar2","Othr",..: 5 5 1 5 5 5 5 5 5 5 ...
##  $ MiscVal              : int  0 0 12500 0 0 0 0 0 0 0 ...
##  $ MoSold               : int  5 6 6 4 3 6 4 1 3 6 ...
##  $ YrSold               : int  2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
##  $ SaleType             : Factor w/ 10 levels "COD","Con","ConLD",..: 10 10 10 10 10 10 10 10 10 10
##  $ SaleCondition        : Factor w/ 6 levels "Abnorml","AdjLand",..: 5 5 5 5 5 5 5 5 5 5 ...
##  $ SalePrice            : int  215000 105000 172000 244000 189900 195500 213500 191500 236500 189000
##  $ MSZoning_RH          : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ MSZoning_FV          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MSZoning_RM          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MSZoning_C (all)     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MSZoning_I (all)     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MSZoning_A (agr)     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Street_Grvl          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Alley_Pave           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Alley_Grvl           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ LotShape_Reg         : int  0 1 0 1 0 0 1 0 0 1 ...
##  $ LotShape_IR2         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ LotShape_IR3         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ LandContour_HLS      : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ LandContour_Bnk      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ LandContour_Low      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Utilities_NoSewr     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Utilities_NoSeWa     : int  0 0 0 0 0 0 0 0 0 0 ...
##   [list output truncated]
##  - attr(*, ".internal.selfref")=<externalptr>
```

The dummies were created in preparation for scaling the features. We don't want to scale the target, *SalePrice*, so we remove that from the main data set and put it into its own object. The dummies are numeric representations of factors, expressed as zeroes and ones. With the dummies created, we don't need the factors in the data set anymore so we remove them. Run a sanity check to make sure all looks good.

```
AmesDumNoTg <- AmesDummy[, -82]
AmesTarget <- AmesDummy[, 82]
names(Filter(is.factor, AmesDumNoTg))
```

```
##  [1] "MSZoning"      "Street"        "Alley"         "LotShape"
##  [5] "LandContour"   "Utilities"     "LotConfig"     "LandSlope"
##  [9] "Neighborhood"  "Condition1"    "Condition2"    "BldgType"
## [13] "HouseStyle"    "RoofStyle"     "RoofMatl"      "Exterior1st"
## [17] "Exterior2nd"   "MasVnrType"    "ExterQual"     "ExterCond"
## [21] "Foundation"    "BsmtQual"      "BsmtCond"      "BsmtExposure"
## [25] "BsmtFinType1"  "BsmtFinType2"  "Heating"       "HeatingQC"
## [29] "CentralAir"    "Electrical"    "BsmtFullBath"  "BsmtHalfBath"
## [33] "KitchenQual"   "Functional"    "FireplaceQu"   "GarageType"
```

```
## [37] "GarageFinish"  "GarageCars"    "GarageQual"    "GarageCond"
## [41] "PavedDrive"    "PoolQC"        "Fence"         "MiscFeature"
## [45] "SaleType"      "SaleCondition"
```

```
dropCols <- c("MSZoning", "Street", "Alley", "LotShape", "LandContour", "Utilities",
              "LotConfig", "LandSlope", "Neighborhood", "Condition1", "Condition2",
              "BldgType", "HouseStyle", "RoofStyle", "RoofMatl", "Exterior1st",
              "Exterior2nd", "MasVnrType", "ExterQual", "ExterCond", "Foundation",
              "BsmtQual", "BsmtCond", "BsmtExposure", "BsmtFinType1", "BsmtFinType2",
              "Heating", "HeatingQC", "CentralAir", "Electrical", "BsmtFullBath",
              "BsmtHalfBath", "KitchenQual", "Functional", "FireplaceQu", "GarageType",
              "GarageFinish", "GarageCars", "GarageQual", "GarageCond", "PavedDrive",
              "PoolQC", "Fence", "MiscFeature", "SaleType", "SaleCondition")
AmesDumNoTg <- AmesDumNoTg %>% dplyr::select(-one_of(dropCols))
dim(AmesDumNoTg)
```

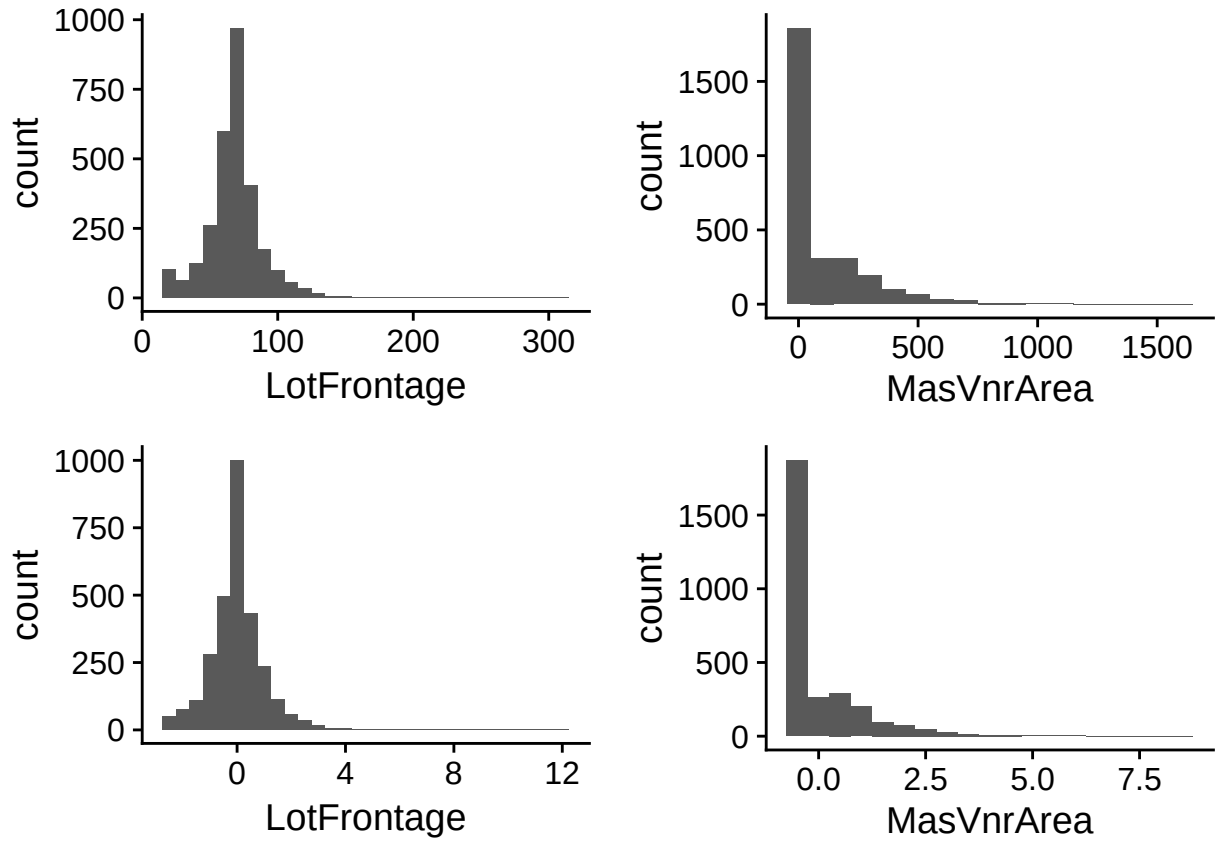
```
## [1] 2925  293
```

Now that our data are in shape and all features are numeric, it is time to scale the data. Once scaled, we need to convert the data back into *tibble* format.

```
AmesDumScale <- scale(AmesDumNoTg, center = TRUE, scale = TRUE)
AmesDumScaleDF <- as.tibble(AmesDumScale)
```

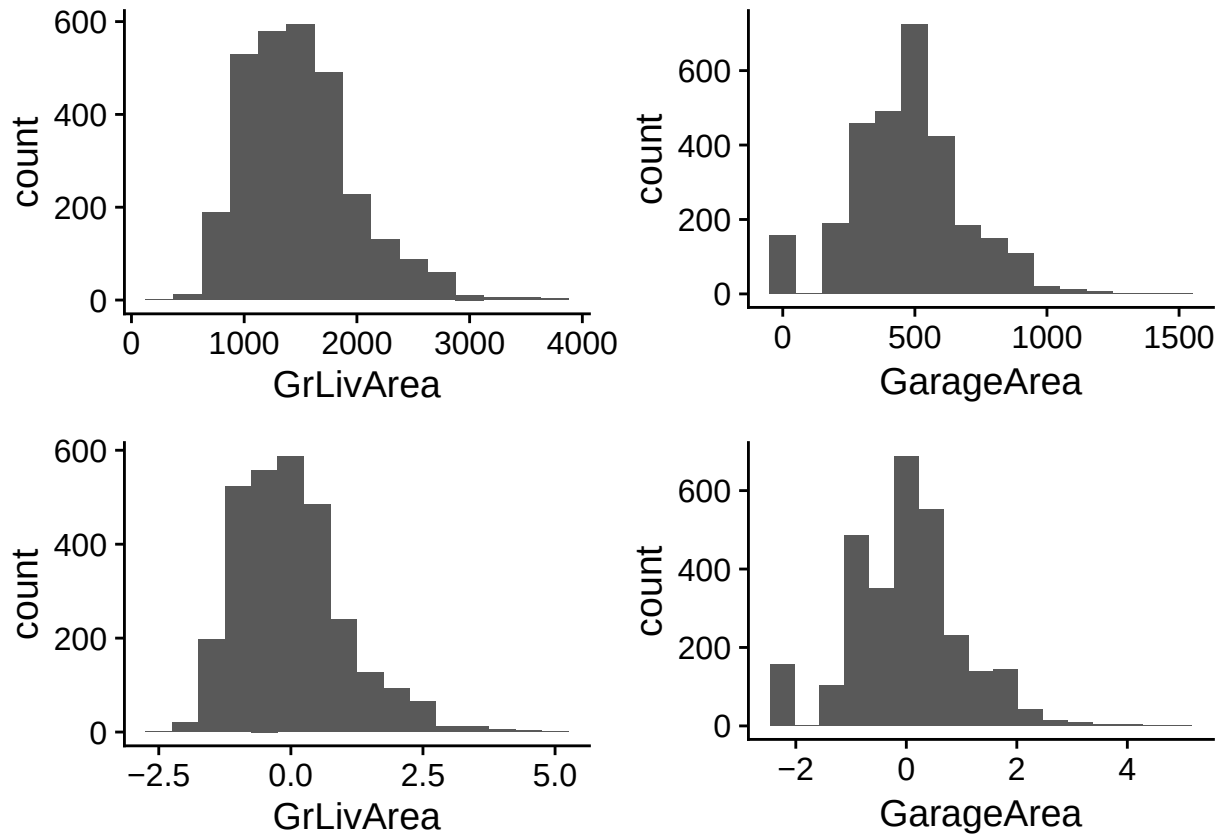
Again, the number of features in this data set is too large to show all transformations, but we can select a few to look at. The point of scaling is that the distributions do not change. All that is impacted is the scale of the x-axis. We can see these characteristics in the plots that follow.

```
p1 <- ggplot(AmesTibble, aes(x = LotFrontage)) + geom_histogram(binwidth = 10)
p2 <- ggplot(AmesDumScaleDF, aes(x = LotFrontage)) + geom_histogram(binwidth = .5)
p3 <- ggplot(AmesTibble, aes(x = MasVnrArea)) + geom_histogram(binwidth = 100)
p4 <- ggplot(AmesDumScaleDF, aes(x = MasVnrArea)) + geom_histogram(binwidth = .5)
p5 <- ggplot(AmesTibble, aes(x = GrLivArea)) + geom_histogram(binwidth = 250)
p6 <- ggplot(AmesDumScaleDF, aes(x = GrLivArea)) + geom_histogram(binwidth = .5)
p7 <- ggplot(AmesTibble, aes(x = GarageArea)) + geom_histogram(binwidth = 100)
p8 <- ggplot(AmesDumScaleDF, aes(x = GarageArea)) + geom_histogram(binwidth = .45)
grid.arrange(p1, p3, p2, p4, nrow = 2)
```



These plots also exhibit no change in distribution, just different x-axis values.

```
grid.arrange(p5, p7, p6, p8, nrow = 2)
```



With the data dummified and scaled, it is time to turn attention to the task of figuring out which of the now nearly 300 features are the most important for the purpose of modelling and predicting *SalePrice*. We need to rejoin the target to the data set in order to do this.

There are a variety of methods we can use to identify candidates for the most important features. This data set is designed for multilinear analysis, so we start with that. But let's take an intuitive guess as to which 10 features are the most important. We can compare the guesstimates to the statistical analysis later.

```
Top10Intuitive <- c('LotArea', 'LotFrontage', 'Neighborhood', 'GrLivArea',
                    'PoolArea', 'X1stFlrSF', 'X2ndFlrSF', 'TotalBsmtSF',
                    'GarageArea', 'KitchenQual')
tibble(Top10Intuitive)
```

```
## # A tibble: 10 x 1
##    Top10Intuitive
##    <chr>
##  1 LotArea
##  2 LotFrontage
##  3 Neighborhood
##  4 GrLivArea
##  5 PoolArea
##  6 X1stFlrSF
##  7 X2ndFlrSF
##  8 TotalBsmtSF
##  9 GarageArea
## 10 KitchenQual
```

```
AmesDumScaleDF <- cbind(AmesTarget, AmesDumScaleDF)
```

A linear model can be built to identify the top 10 features, based on the value of the estimates each feature is assigned by the model.

```
lmRating <- lm(SalePrice ~ . , AmesDumScaleDF)
lmCoefs <- tidy(lmRating)
lmCoefs$abs_est <- sqrt(lmCoefs$estimate^2)
lmCoefsAbs <- lmCoefs[-c(2:5)]
lmCoefsAbs <- lmCoefsAbs[-1,]
head(lmCoefsAbs[order(-lmCoefsAbs$abs_est), ], 10)
```

```
##                    term    abs_est
## 15          TotalBsmtSF 123574.857
## 14            BsmtUnfSF 120415.319
## 12           BsmtFinSF1 111278.962
## 13           BsmtFinSF2  44059.849
## 17            X2ndFlrSF  26468.549
## 16            X1stFlrSF  20332.979
## 123 Exterior1st_VinylSd  11834.644
## 32             PoolArea   9796.983
## 9             YearBuilt   9606.780
## 7           OverallQual   8777.383
```

The top 10 features are listed above, but let's use the same method to identify the least important features, as well.

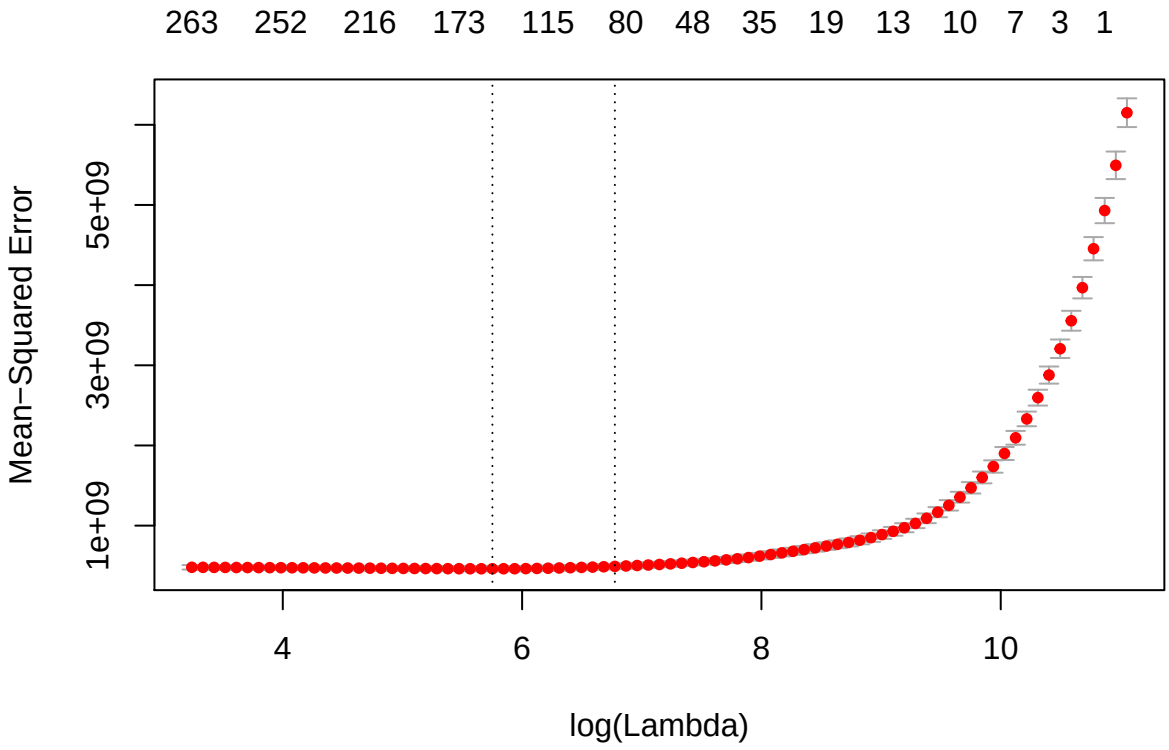
```
tail(lmCoefsAbs[order(-lmCoefsAbs$abs_est), ], 10)
```

```
##                   term   abs_est
## 230 GarageType_Basment 50.570919
## 91     Condition1_PosA 40.573687
## 203   Electrical_FuseA 38.236848
## 108    HouseStyle_SLvl 34.245808
## 30          X3SsnPorch 22.521423
## 205   Electrical_FuseP 11.404369
## 211     BsmtHalfBath_1  9.120788
## 262   MiscFeature_Shed  8.161129
## 121      RoofMatl_Roll  7.673177
## 100    Condition2_RRAn  2.343051
```

Apparently, the electrical fuse box is not a consideration when buying a home or determining its price.

Next, lassos will be employed to identify the top 10 features. The first effort will use a Gaussian distribution. The aim here is to minimize lambda using cross validation. The `cv.glmnet()` command uses the squared-error as the loss function. We can plot the values of lambda as they minimize across the features.

```
x <- as.matrix(AmesDumScaleDF[, -1])
y <- as.matrix(AmesDumScaleDF[, 1])
set.seed(805)
cv.ridge.g <- cv.glmnet(x, y, family = 'gaussian')
plot(cv.ridge.g)
```



Running through the analysis yields the top 10 features the Guassian lasso identifies. The bottom 10 are also included.

```
cv.ridge.g$lambda.min
```

```
## [1] 314.7688
```

```
cv.ridge.g$lambda.1se
```

```
## [1] 875.8628
```

```
LassoCoefsG <- coef(cv.ridge.g, s = cv.ridge.g$lambda.min)
LassoCoefsG <- tidy(LassoCoefsG)
LassoCoefsG$abs_est <- sqrt(LassoCoefsG$value^2)
LassoCoefsG <- LassoCoefsG[-c(2:3)]
LassoCoefsG <- LassoCoefsG[-1,]
head(LassoCoefsG[order(-LassoCoefsG$abs_est), ], 10)
```

```
##                  row   abs_est
## 14         GrLivArea 27445.891
## 5        OverallQual 10817.574
## 7          YearBuilt  8060.114
## 10        BsmtFinSF1  7882.249
## 12       TotalBsmtSF  7648.576
## 6        OverallCond  5589.734
## 101 BsmtExposure_No  5442.511
## 141    GarageCars_3  5406.970
## 93     ExterQual_Ex  4983.285
```

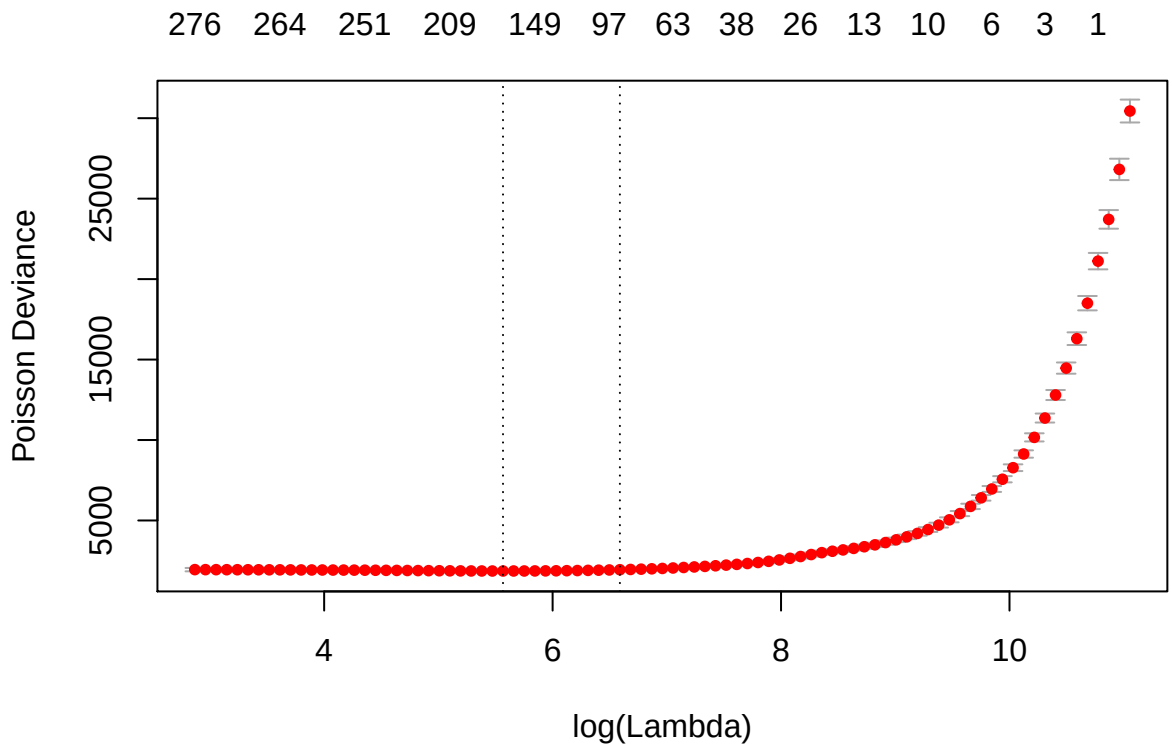
```
## 99      BsmtQual_Ex  4874.542
```

```
tail(LassoCoefsG[order(-LassoCoefsG$abs_est), ], 10)
```

```
##                       row    abs_est
## 136       FireplaceQu_Fa 22.7398430
## 138   GarageType_CarPort 15.5561731
## 44   Neighborhood_BrDale 13.8200880
## 109     BsmtFinType2_BLQ 11.9114936
## 94          ExterQual_Fa 10.3457489
## 85   Exterior2nd_CmentBd  8.7584850
## 156 SaleCondition_Alloca  8.6983484
## 153         SaleType_CWD  3.5121652
## 87   Exterior2nd_PreCast  1.5589606
## 78   Exterior1st_CemntBd  0.7499508
```

Next, we will use a lasso with a Poisson distribution to see if it yields different results. Here, the loss fucntion is represented by deviance. Top and bottom features are displayed.

```
set.seed(805)
cv.ridge.p <- cv.glmnet(x, y, family = 'poisson')
plot(cv.ridge.p)
```



Running through the analysis identifies the top 10 features for the Poisson lasso.

```
cv.ridge.p$lambda.min
```

```
## [1] 261.3266
```

```
cv.ridge.p$lambda.1se
```

```
## [1] 727.1567
```

```
LassoCoefsP <- coef(cv.ridge.p, s = cv.ridge.p$lambda.min)
LassoCoefsP <- tidy(LassoCoefsP)
LassoCoefsP$abs_est <- sqrt(LassoCoefsP$value^2)
LassoCoefsP <- LassoCoefsP[-c(2:3)]
LassoCoefsP <- LassoCoefsP[-1,]
head(LassoCoefsP[order(-LassoCoefsP$abs_est), ], 10)
```

```
##                     row    abs_est
## 17            GrLivArea 0.12731172
## 7           OverallQual 0.07689086
## 9             YearBuilt 0.05551847
## 14          TotalBsmtSF 0.04153414
## 8           OverallCond 0.03968042
## 12           BsmtFinSF1 0.03090506
## 24           GarageArea 0.02125383
## 57 Neighborhood_Crawfor 0.02015258
## 31          MSZoning_RM 0.01755636
## 10         YearRemodAdd 0.01688563
```

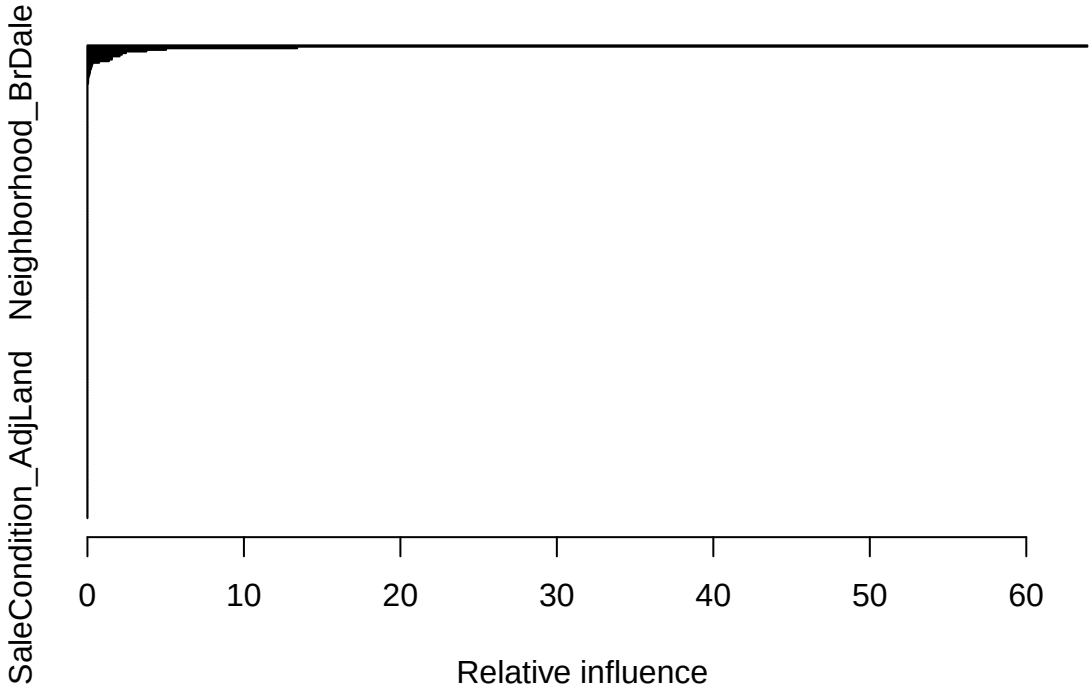
```
tail(LassoCoefsP[order(-LassoCoefsP$abs_est), ], 10)
```

```
##                     row      abs_est
## 151        GarageCars_4 3.962979e-05
## 86  Exterior2nd_Wd Shng 3.513499e-05
## 111    BsmtFinType1_w/o 1.931424e-05
## 155       GarageQual_Po 1.570485e-05
## 156      GarageCond_w/o 1.036835e-05
## 126      BsmtFullBath_2 1.035090e-05
## 150        GarageCars_0 9.358394e-06
## 114    BsmtFinType2_w/o 8.298147e-06
## 37         LotShape_IR3 6.553659e-06
## 129    BsmtHalfBath_w/o 7.497908e-17
```

Of the two lasso procedures, it appears the Poisson distribution is most useful because its lambda values are lower.

Next, we will use a Gradient Boost Method for the same purpose.

```
AmesBoost <- gbm(SalePrice ~ . , data = AmesDumScaleDF, distribution = 'gaussian',
                 n.trees = 100, shrinkage = 0.01, interaction.depth = 4)
summary(AmesBoost)
```




```
##                                      var     rel.inf
## OverallQual                   OverallQual 63.90282548
## GrLivArea                       GrLivArea 13.40399035
## GarageCars_3                 GarageCars_3  5.02478512
## TotalBsmtSF                   TotalBsmtSF  3.77016254
## BsmtQual_Ex                   BsmtQual_Ex  2.46371610
## X1stFlrSF                       X1stFlrSF  2.20032001
## YearBuilt                       YearBuilt  2.05638089
## FullBath                         FullBath  1.56479915
## GarageArea                     GarageArea  1.56196647
## BsmtFinSF1                     BsmtFinSF1  1.37285135
## KitchenQual_Ex             KitchenQual_Ex  0.74520205
## LotArea                           LotArea  0.31820903
## ExterQual_Ex                 ExterQual_Ex  0.28286206
## MasVnrArea                     MasVnrArea  0.25490636
## X2ndFlrSF                       X2ndFlrSF  0.22180169
## YearRemodAdd                 YearRemodAdd  0.17593074
## Fireplaces                     Fireplaces  0.16488555
## KitchenQual_Gd             KitchenQual_Gd  0.13942962
## `FireplaceQu_w/o`       `FireplaceQu_w/o`  0.11327372
## ExterQual_Gd                 ExterQual_Gd  0.07676505
## BsmtQual_Gd                   BsmtQual_Gd  0.05163028
## CentralAir_N                 CentralAir_N  0.04718140
## GarageType_Detchd       GarageType_Detchd  0.04702996
## MSZoning_RM                   MSZoning_RM  0.03909503
## Order                               Order  0.00000000
```

```
## PID                                PID  0.00000000
## MSSubClass                  MSSubClass  0.00000000
## LotFrontage                LotFrontage  0.00000000
## OverallCond                OverallCond  0.00000000
## BsmtFinSF2                  BsmtFinSF2  0.00000000
## BsmtUnfSF                    BsmtUnfSF  0.00000000
## LowQualFinSF              LowQualFinSF  0.00000000
## HalfBath                     HalfBath  0.00000000
## BedroomAbvGr              BedroomAbvGr  0.00000000
## KitchenAbvGr              KitchenAbvGr  0.00000000
## TotRmsAbvGrd              TotRmsAbvGrd  0.00000000
## GarageYrBlt                GarageYrBlt  0.00000000
## WoodDeckSF                  WoodDeckSF  0.00000000
## OpenPorchSF                OpenPorchSF  0.00000000
## EnclosedPorch            EnclosedPorch  0.00000000
## X3SsnPorch                   X3SsnPorch  0.00000000
## ScreenPorch                ScreenPorch  0.00000000
## PoolArea                      PoolArea  0.00000000
## MiscVal                        MiscVal  0.00000000
## MoSold                          MoSold  0.00000000
## YrSold                          YrSold  0.00000000
## MSZoning_RH                MSZoning_RH  0.00000000
## MSZoning_FV                MSZoning_FV  0.00000000
## `MSZoning_C (all)`  `MSZoning_C (all)`  0.00000000
## `MSZoning_I (all)`  `MSZoning_I (all)`  0.00000000
## `MSZoning_A (agr)`  `MSZoning_A (agr)`  0.00000000
## Street_Grvl                Street_Grvl  0.00000000
## Alley_Pave                  Alley_Pave  0.00000000
## Alley_Grvl                  Alley_Grvl  0.00000000
## LotShape_Reg              LotShape_Reg  0.00000000
## LotShape_IR2              LotShape_IR2  0.00000000
## LotShape_IR3              LotShape_IR3  0.00000000
## LandContour_HLS        LandContour_HLS  0.00000000
## LandContour_Bnk        LandContour_Bnk  0.00000000
## LandContour_Low        LandContour_Low  0.00000000
## Utilities_NoSewr      Utilities_NoSewr  0.00000000
## Utilities_NoSeWa      Utilities_NoSeWa  0.00000000
## LotConfig_Inside      LotConfig_Inside  0.00000000
## LotConfig_CulDSac    LotConfig_CulDSac  0.00000000
## LotConfig_FR2            LotConfig_FR2  0.00000000
## LotConfig_FR3            LotConfig_FR3  0.00000000
## LandSlope_Mod            LandSlope_Mod  0.00000000
## LandSlope_Sev            LandSlope_Sev  0.00000000
## Neighborhood_Gilbert  Neighborhood_Gilbert  0.00000000
## Neighborhood_StoneBr  Neighborhood_StoneBr  0.00000000
## Neighborhood_NWAmes    Neighborhood_NWAmes  0.00000000
## Neighborhood_Somerst  Neighborhood_Somerst  0.00000000
## Neighborhood_BrDale    Neighborhood_BrDale  0.00000000
## Neighborhood_NPkVill  Neighborhood_NPkVill  0.00000000
## Neighborhood_NridgHt  Neighborhood_NridgHt  0.00000000
## Neighborhood_Blmngtn  Neighborhood_Blmngtn  0.00000000
## Neighborhood_NoRidge  Neighborhood_NoRidge  0.00000000
## Neighborhood_SawyerW  Neighborhood_SawyerW  0.00000000
## Neighborhood_Sawyer    Neighborhood_Sawyer  0.00000000
```

```
## Neighborhood_Greens     Neighborhood_Greens  0.00000000
## Neighborhood_BrkSide   Neighborhood_BrkSide  0.00000000
## Neighborhood_OldTown   Neighborhood_OldTown  0.00000000
## Neighborhood_IDOTRR     Neighborhood_IDOTRR  0.00000000
## Neighborhood_ClearCr   Neighborhood_ClearCr  0.00000000
## Neighborhood_SWISU       Neighborhood_SWISU  0.00000000
## Neighborhood_Edwards   Neighborhood_Edwards  0.00000000
## Neighborhood_CollgCr   Neighborhood_CollgCr  0.00000000
## Neighborhood_Crawfor   Neighborhood_Crawfor  0.00000000
## Neighborhood_Blueste   Neighborhood_Blueste  0.00000000
## Neighborhood_Mitchel   Neighborhood_Mitchel  0.00000000
## Neighborhood_Timber     Neighborhood_Timber  0.00000000
## Neighborhood_MeadowV   Neighborhood_MeadowV  0.00000000
## Neighborhood_Veenker   Neighborhood_Veenker  0.00000000
## Neighborhood_GrnHill   Neighborhood_GrnHill  0.00000000
## Neighborhood_Landmrk   Neighborhood_Landmrk  0.00000000
## Condition1_Feedr           Condition1_Feedr  0.00000000
## Condition1_PosN             Condition1_PosN  0.00000000
## Condition1_RRNe             Condition1_RRNe  0.00000000
## Condition1_RRAe             Condition1_RRAe  0.00000000
## Condition1_Artery         Condition1_Artery  0.00000000
## Condition1_PosA             Condition1_PosA  0.00000000
## Condition1_RRAn             Condition1_RRAn  0.00000000
## Condition1_RRNn             Condition1_RRNn  0.00000000
## Condition2_Feedr           Condition2_Feedr  0.00000000
## Condition2_PosA             Condition2_PosA  0.00000000
## Condition2_PosN             Condition2_PosN  0.00000000
## Condition2_Artery         Condition2_Artery  0.00000000
## Condition2_RRNn             Condition2_RRNn  0.00000000
## Condition2_RRAe             Condition2_RRAe  0.00000000
## Condition2_RRAn             Condition2_RRAn  0.00000000
## BldgType_TwnhsE             BldgType_TwnhsE  0.00000000
## BldgType_Twnhs               BldgType_Twnhs  0.00000000
## BldgType_Duplex             BldgType_Duplex  0.00000000
## BldgType_2fmCon             BldgType_2fmCon  0.00000000
## HouseStyle_2Story         HouseStyle_2Story  0.00000000
## HouseStyle_1.5Fin         HouseStyle_1.5Fin  0.00000000
## HouseStyle_SFoyer         HouseStyle_SFoyer  0.00000000
## HouseStyle_SLvl             HouseStyle_SLvl  0.00000000
## HouseStyle_2.5Unf         HouseStyle_2.5Unf  0.00000000
## HouseStyle_1.5Unf         HouseStyle_1.5Unf  0.00000000
## HouseStyle_2.5Fin         HouseStyle_2.5Fin  0.00000000
## RoofStyle_Gable             RoofStyle_Gable  0.00000000
## RoofStyle_Mansard         RoofStyle_Mansard  0.00000000
## RoofStyle_Gambrel         RoofStyle_Gambrel  0.00000000
## RoofStyle_Shed               RoofStyle_Shed  0.00000000
## RoofStyle_Flat               RoofStyle_Flat  0.00000000
## RoofMatl_WdShake           RoofMatl_WdShake  0.00000000
## `RoofMatl_Tar&Grv`       `RoofMatl_Tar&Grv`  0.00000000
## RoofMatl_WdShngl           RoofMatl_WdShngl  0.00000000
## RoofMatl_Membran           RoofMatl_Membran  0.00000000
## RoofMatl_Roll                 RoofMatl_Roll  0.00000000
## RoofMatl_Metal               RoofMatl_Metal  0.00000000
## Exterior1st_VinylSd     Exterior1st_VinylSd  0.00000000
```

```
## `Exterior1st_Wd Sdng` `Exterior1st_Wd Sdng`  0.00000000
## Exterior1st_CemntBd     Exterior1st_CemntBd  0.00000000
## Exterior1st_HdBoard     Exterior1st_HdBoard  0.00000000
## Exterior1st_Plywood     Exterior1st_Plywood  0.00000000
## Exterior1st_MetalSd     Exterior1st_MetalSd  0.00000000
## Exterior1st_AsbShng     Exterior1st_AsbShng  0.00000000
## Exterior1st_WdShing     Exterior1st_WdShing  0.00000000
## Exterior1st_Stucco       Exterior1st_Stucco  0.00000000
## Exterior1st_AsphShn     Exterior1st_AsphShn  0.00000000
## Exterior1st_BrkComm     Exterior1st_BrkComm  0.00000000
## Exterior1st_CBlock       Exterior1st_CBlock  0.00000000
## Exterior1st_PreCast     Exterior1st_PreCast  0.00000000
## Exterior1st_Stone         Exterior1st_Stone  0.00000000
## Exterior1st_ImStucc     Exterior1st_ImStucc  0.00000000
## Exterior2nd_VinylSd     Exterior2nd_VinylSd  0.00000000
## `Exterior2nd_Wd Sdng` `Exterior2nd_Wd Sdng`  0.00000000
## Exterior2nd_BrkFace     Exterior2nd_BrkFace  0.00000000
## Exterior2nd_CmentBd     Exterior2nd_CmentBd  0.00000000
## Exterior2nd_HdBoard     Exterior2nd_HdBoard  0.00000000
## `Exterior2nd_Wd Shng` `Exterior2nd_Wd Shng`  0.00000000
## Exterior2nd_MetalSd     Exterior2nd_MetalSd  0.00000000
## Exterior2nd_ImStucc     Exterior2nd_ImStucc  0.00000000
## `Exterior2nd_Brk Cmn` `Exterior2nd_Brk Cmn`  0.00000000
## Exterior2nd_AsbShng     Exterior2nd_AsbShng  0.00000000
## Exterior2nd_Stucco       Exterior2nd_Stucco  0.00000000
## Exterior2nd_AsphShn     Exterior2nd_AsphShn  0.00000000
## Exterior2nd_CBlock       Exterior2nd_CBlock  0.00000000
## Exterior2nd_Stone         Exterior2nd_Stone  0.00000000
## Exterior2nd_PreCast     Exterior2nd_PreCast  0.00000000
## Exterior2nd_Other         Exterior2nd_Other  0.00000000
## MasVnrType_None             MasVnrType_None  0.00000000
## MasVnrType_BrkFace       MasVnrType_BrkFace  0.00000000
## MasVnrType_BrkCmn         MasVnrType_BrkCmn  0.00000000
## MasVnrType_CBlock         MasVnrType_CBlock  0.00000000
## ExterQual_Fa                   ExterQual_Fa  0.00000000
## ExterCond_Gd                   ExterCond_Gd  0.00000000
## ExterCond_Fa                   ExterCond_Fa  0.00000000
## ExterCond_Po                   ExterCond_Po  0.00000000
## ExterCond_Ex                   ExterCond_Ex  0.00000000
## Foundation_PConc           Foundation_PConc  0.00000000
## Foundation_Wood             Foundation_Wood  0.00000000
## Foundation_BrkTil         Foundation_BrkTil  0.00000000
## Foundation_Slab             Foundation_Slab  0.00000000
## Foundation_Stone           Foundation_Stone  0.00000000
## `BsmtQual_w/o`               `BsmtQual_w/o`  0.00000000
## BsmtQual_Fa                     BsmtQual_Fa  0.00000000
## BsmtQual_                         BsmtQual_  0.00000000
## BsmtQual_Po                     BsmtQual_Po  0.00000000
## BsmtCond_TA                     BsmtCond_TA  0.00000000
## `BsmtCond_w/o`               `BsmtCond_w/o`  0.00000000
## BsmtCond_Po                     BsmtCond_Po  0.00000000
## BsmtCond_Fa                     BsmtCond_Fa  0.00000000
## BsmtCond_Ex                     BsmtCond_Ex  0.00000000
## BsmtCond_                         BsmtCond_  0.00000000
```

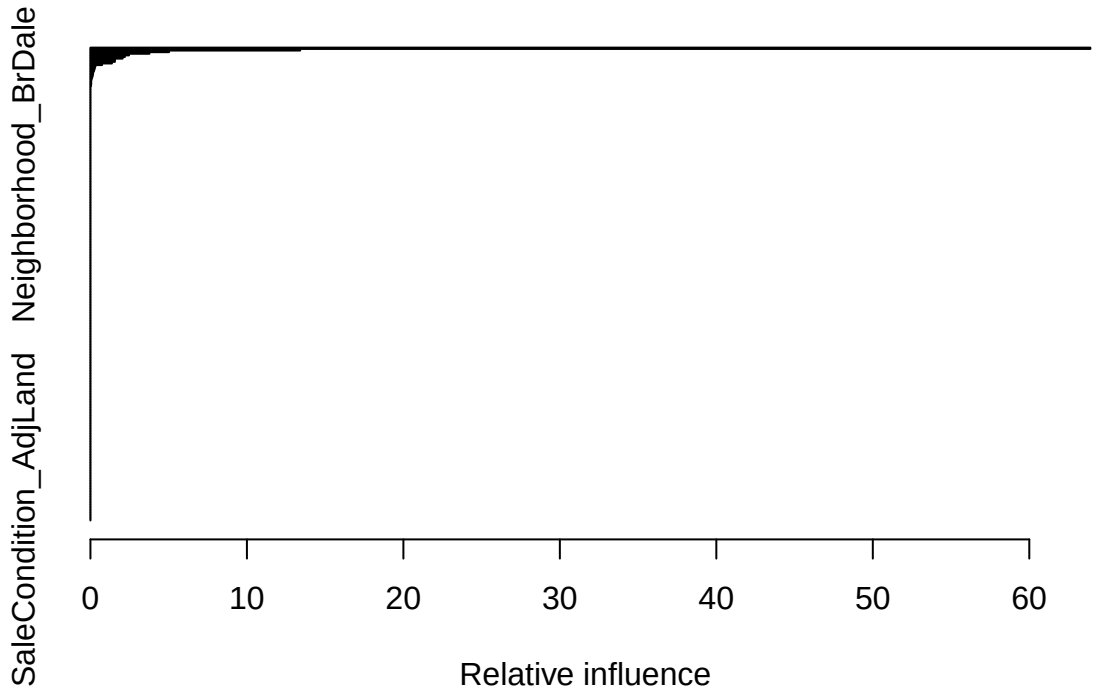
```
## BsmtExposure_No            BsmtExposure_No  0.00000000
## BsmtExposure_Mn            BsmtExposure_Mn  0.00000000
## BsmtExposure_Av            BsmtExposure_Av  0.00000000
## BsmtExposure_                BsmtExposure_  0.00000000
## `BsmtExposure_w/o`       `BsmtExposure_w/o`  0.00000000
## BsmtFinType1_Rec           BsmtFinType1_Rec  0.00000000
## BsmtFinType1_ALQ           BsmtFinType1_ALQ  0.00000000
## BsmtFinType1_GLQ           BsmtFinType1_GLQ  0.00000000
## BsmtFinType1_Unf           BsmtFinType1_Unf  0.00000000
## BsmtFinType1_LwQ           BsmtFinType1_LwQ  0.00000000
## `BsmtFinType1_w/o`       `BsmtFinType1_w/o`  0.00000000
## BsmtFinType1_                BsmtFinType1_  0.00000000
## BsmtFinType2_LwQ           BsmtFinType2_LwQ  0.00000000
## BsmtFinType2_BLQ           BsmtFinType2_BLQ  0.00000000
## BsmtFinType2_Rec           BsmtFinType2_Rec  0.00000000
## `BsmtFinType2_w/o`       `BsmtFinType2_w/o`  0.00000000
## BsmtFinType2_GLQ           BsmtFinType2_GLQ  0.00000000
## BsmtFinType2_ALQ           BsmtFinType2_ALQ  0.00000000
## BsmtFinType2_                BsmtFinType2_  0.00000000
## Heating_GasW                  Heating_GasW  0.00000000
## Heating_Grav                  Heating_Grav  0.00000000
## Heating_Wall                  Heating_Wall  0.00000000
## Heating_Floor                Heating_Floor  0.00000000
## Heating_OthW                  Heating_OthW  0.00000000
## HeatingQC_TA                  HeatingQC_TA  0.00000000
## HeatingQC_Ex                  HeatingQC_Ex  0.00000000
## HeatingQC_Gd                  HeatingQC_Gd  0.00000000
## HeatingQC_Po                  HeatingQC_Po  0.00000000
## Electrical_FuseA           Electrical_FuseA  0.00000000
## Electrical_FuseF           Electrical_FuseF  0.00000000
## Electrical_FuseP           Electrical_FuseP  0.00000000
## Electrical_Mix               Electrical_Mix  0.00000000
## BsmtFullBath_0               BsmtFullBath_0  0.00000000
## BsmtFullBath_2               BsmtFullBath_2  0.00000000
## BsmtFullBath_3               BsmtFullBath_3  0.00000000
## `BsmtFullBath_w/o`       `BsmtFullBath_w/o`  0.00000000
## BsmtHalfBath_1               BsmtHalfBath_1  0.00000000
## `BsmtHalfBath_w/o`       `BsmtHalfBath_w/o`  0.00000000
## BsmtHalfBath_2               BsmtHalfBath_2  0.00000000
## KitchenQual_Fa               KitchenQual_Fa  0.00000000
## KitchenQual_Po               KitchenQual_Po  0.00000000
## Functional_Mod               Functional_Mod  0.00000000
## Functional_Min1             Functional_Min1  0.00000000
## Functional_Min2             Functional_Min2  0.00000000
## Functional_Maj1             Functional_Maj1  0.00000000
## Functional_Maj2             Functional_Maj2  0.00000000
## Functional_Sev               Functional_Sev  0.00000000
## Functional_Sal               Functional_Sal  0.00000000
## FireplaceQu_TA               FireplaceQu_TA  0.00000000
## FireplaceQu_Po               FireplaceQu_Po  0.00000000
## FireplaceQu_Ex               FireplaceQu_Ex  0.00000000
## FireplaceQu_Fa               FireplaceQu_Fa  0.00000000
## GarageType_BuiltIn       GarageType_BuiltIn  0.00000000
## GarageType_Basment       GarageType_Basment  0.00000000
```

```
## `GarageType_w/o`           `GarageType_w/o`  0.00000000
## GarageType_CarPort       GarageType_CarPort  0.00000000
## GarageType_2Types         GarageType_2Types  0.00000000
## GarageFinish_Unf           GarageFinish_Unf  0.00000000
## GarageFinish_RFn           GarageFinish_RFn  0.00000000
## `GarageFinish_w/o`       `GarageFinish_w/o`  0.00000000
## GarageFinish_                 GarageFinish_  0.00000000
## GarageCars_1                   GarageCars_1  0.00000000
## GarageCars_0                   GarageCars_0  0.00000000
## GarageCars_4                   GarageCars_4  0.00000000
## GarageCars_5                   GarageCars_5  0.00000000
## `GarageCars_w/o`           `GarageCars_w/o`  0.00000000
## `GarageQual_w/o`           `GarageQual_w/o`  0.00000000
## GarageQual_Fa                 GarageQual_Fa  0.00000000
## GarageQual_Gd                 GarageQual_Gd  0.00000000
## GarageQual_Ex                 GarageQual_Ex  0.00000000
## GarageQual_Po                 GarageQual_Po  0.00000000
## GarageQual_                     GarageQual_  0.00000000
## `GarageCond_w/o`           `GarageCond_w/o`  0.00000000
## GarageCond_Fa                 GarageCond_Fa  0.00000000
## GarageCond_Gd                 GarageCond_Gd  0.00000000
## GarageCond_Ex                 GarageCond_Ex  0.00000000
## GarageCond_Po                 GarageCond_Po  0.00000000
## GarageCond_                     GarageCond_  0.00000000
## PavedDrive_Y                   PavedDrive_Y  0.00000000
## PavedDrive_N                   PavedDrive_N  0.00000000
## PoolQC_Ex                         PoolQC_Ex  0.00000000
## PoolQC_Gd                         PoolQC_Gd  0.00000000
## PoolQC_TA                         PoolQC_TA  0.00000000
## PoolQC_Fa                         PoolQC_Fa  0.00000000
## Fence_MnPrv                     Fence_MnPrv  0.00000000
## Fence_GdPrv                     Fence_GdPrv  0.00000000
## Fence_GdWo                       Fence_GdWo  0.00000000
## Fence_MnWw                       Fence_MnWw  0.00000000
## MiscFeature_Gar2           MiscFeature_Gar2  0.00000000
## MiscFeature_Shed           MiscFeature_Shed  0.00000000
## MiscFeature_Othr           MiscFeature_Othr  0.00000000
## MiscFeature_TenC           MiscFeature_TenC  0.00000000
## SaleType_New                   SaleType_New  0.00000000
## SaleType_COD                   SaleType_COD  0.00000000
## SaleType_ConLI               SaleType_ConLI  0.00000000
## SaleType_Con                   SaleType_Con  0.00000000
## SaleType_ConLD               SaleType_ConLD  0.00000000
## SaleType_Oth                   SaleType_Oth  0.00000000
## SaleType_ConLw               SaleType_ConLw  0.00000000
## SaleType_CWD                   SaleType_CWD  0.00000000
## SaleType_VWD                   SaleType_VWD  0.00000000
## SaleCondition_Partial SaleCondition_Partial  0.00000000
## SaleCondition_Family   SaleCondition_Family  0.00000000
## SaleCondition_Abnorml SaleCondition_Abnorml  0.00000000
## SaleCondition_Alloca   SaleCondition_Alloca  0.00000000
## SaleCondition_AdjLand SaleCondition_AdjLand  0.00000000
```

We can identify the top 10 features easily enough. The graph is not too useful because of nearly 300 features,

the GBM considers only 24, less than 10%, worth consideration.

```
head(summary(AmesBoost), 10)
```




```
##                         var   rel.inf
## OverallQual   OverallQual 63.902825
## GrLivArea       GrLivArea 13.403990
## GarageCars_3 GarageCars_3  5.024785
## TotalBsmtSF   TotalBsmtSF  3.770163
## BsmtQual_Ex   BsmtQual_Ex  2.463716
## X1stFlrSF       X1stFlrSF  2.200320
## YearBuilt       YearBuilt  2.056381
## FullBath         FullBath  1.564799
## GarageArea     GarageArea  1.561966
## BsmtFinSF1     BsmtFinSF1  1.372851
```

A table in the *README.md* file compares the results of the five feature selection tests.

The selection rule for features was this: if any of the top 10 features is selected by at least two of the five methods, it will be included in the Principal Components Analysis. As it happens, 11 features passed the test. So we create a new *tibble* with just these features. The PCA will determine which 10 combinations of these 11 features are most important to our analysis going forward.

Principal Components Analysis

We want to identify the principal components (PCs) of these top 11 features because we need to how to build out the multilinear regression model we will eventually run to predict *SalePrice*. We are going to run a

principal components regression, in other words, on the data. The PCs we identify will become the variables in the regression model.

A PC is defined by Chris Albon (@chrisalbon on Twitter) as "the linear combination of features that have the maximum variance out of all linear combinations."

The first PC (PC1) is basically the line of best fit for the data we are dealing with. The direction of PC1 indicates the line along which the data vary the most. As will be seen, PC1 in this analysis consists of negative values for all features. This suggests that each of these features is below average in terms of PC1, when compared to corresponding values in subsequent PCs. These relationships will be depicted later graphically.

PC2 is the linear combination of the same 11 features that is uncorrelated to PC1. Because of this, the direction of PC2 is perpendicular to the direction of PC1. PC2 is said to be "orthogonal" to PC1. Because PC1 and PC2 are orthogonal, they contain more information combined than either one does individually. The way PCA works, PC1 contains the most information.

We continue to add PCs to the mix, building on orthogonal relationships between the 11 features, until we find the right number that contains the most information about the data set without overcomplicating the model. Each subsequent PC is uncorrelated (or orthogonal) to the preceding PC and it conveys less information than its immediate predecessor.

Of course, the effectiveness of this approach depends on two crucial assumptions. First, we assume the features we have chosen from the larger data set are in fact the most appropriate features to use. Second, we assume that a relatively small set of PCs can explain most of the data. In other words, we assume the directions associated with our PCs are the same directions associated with our target, which is *SalePrice* in the *Ames* case.

To begin, we create a data frame consisting of the 11 features chosen earlier.

```
AmesPCAMeths <- as.tibble(AmesDumScaleDF[, c(15, 12, 17, 16, 33, 9, 19, 7, 250, 172, 27)])
str(AmesPCAMeths)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    2925 obs. of  11 variables:
##  $ TotalBsmtSF : atomic  0.0788 -0.3915 0.6702 2.5251 -0.2822 ...
##   ..- attr(*, "scaled:center")= Named num  1.46e+03 7.14e+08 5.74e+01 6.89e+01 1.01e+04 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##   ..- attr(*, "scaled:scale")= Named num  8.46e+02 1.89e+08 4.27e+01 2.07e+01 7.78e+03 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##  $ BsmtFinSF1  : atomic  0.4588 0.0683 1.1072 1.4315 0.8058 ...
##   ..- attr(*, "scaled:center")= Named num  1.46e+03 7.14e+08 5.74e+01 6.89e+01 1.01e+04 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##   ..- attr(*, "scaled:scale")= Named num  8.46e+02 1.89e+08 4.27e+01 2.07e+01 7.78e+03 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##  $ X2ndFlrSF   : atomic  -0.784 -0.784 -0.784 -0.784 0.862 ...
##   ..- attr(*, "scaled:center")= Named num  1.46e+03 7.14e+08 5.74e+01 6.89e+01 1.01e+04 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##   ..- attr(*, "scaled:scale")= Named num  8.46e+02 1.89e+08 4.27e+01 2.07e+01 7.78e+03 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##  $ X1stFlrSF   : atomic  1.329 -0.689 0.461 2.535 -0.604 ...
##   ..- attr(*, "scaled:center")= Named num  1.46e+03 7.14e+08 5.74e+01 6.89e+01 1.01e+04 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##   ..- attr(*, "scaled:scale")= Named num  8.46e+02 1.89e+08 4.27e+01 2.07e+01 7.78e+03 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##  $ PoolArea    : atomic  -0.0574 -0.0574 -0.0574 -0.0574 -0.0574 ...
##   ..- attr(*, "scaled:center")= Named num  1.46e+03 7.14e+08 5.74e+01 6.89e+01 1.01e+04 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##   ..- attr(*, "scaled:scale")= Named num  8.46e+02 1.89e+08 4.27e+01 2.07e+01 7.78e+03 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
```

```
##  $ YearBuilt   : atomic  -0.374 -0.341 -0.44 -0.109 0.85 ...
##   ..- attr(*, "scaled:center")= Named num  1.46e+03 7.14e+08 5.74e+01 6.89e+01 1.01e+04 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##   ..- attr(*, "scaled:scale")= Named num  8.46e+02 1.89e+08 4.27e+01 2.07e+01 7.78e+03 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##  $ GrLivArea   : atomic  0.333 -1.23 -0.339 1.267 0.278 ...
##   ..- attr(*, "scaled:center")= Named num  1.46e+03 7.14e+08 5.74e+01 6.89e+01 1.01e+04 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##   ..- attr(*, "scaled:scale")= Named num  8.46e+02 1.89e+08 4.27e+01 2.07e+01 7.78e+03 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##  $ OverallQual : atomic  -0.0629 -0.7757 -0.0629 0.6499 -0.7757 ...
##   ..- attr(*, "scaled:center")= Named num  1.46e+03 7.14e+08 5.74e+01 6.89e+01 1.01e+04 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##   ..- attr(*, "scaled:scale")= Named num  8.46e+02 1.89e+08 4.27e+01 2.07e+01 7.78e+03 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##  $ GarageCars_3: atomic  -0.38 -0.38 -0.38 -0.38 -0.38 ...
##   ..- attr(*, "scaled:center")= Named num  1.46e+03 7.14e+08 5.74e+01 6.89e+01 1.01e+04 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##   ..- attr(*, "scaled:scale")= Named num  8.46e+02 1.89e+08 4.27e+01 2.07e+01 7.78e+03 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##  $ BsmtQual_Ex : atomic  -0.308 -0.308 -0.308 -0.308 -0.308 ...
##   ..- attr(*, "scaled:center")= Named num  1.46e+03 7.14e+08 5.74e+01 6.89e+01 1.01e+04 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##   ..- attr(*, "scaled:scale")= Named num  8.46e+02 1.89e+08 4.27e+01 2.07e+01 7.78e+03 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##  $ GarageArea  : atomic  0.2625 1.2074 -0.7479 0.2344 0.0473 ...
##   ..- attr(*, "scaled:center")= Named num  1.46e+03 7.14e+08 5.74e+01 6.89e+01 1.01e+04 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
##   ..- attr(*, "scaled:scale")= Named num  8.46e+02 1.89e+08 4.27e+01 2.07e+01 7.78e+03 ...
##   .. ..- attr(*, "names")= chr  "Order" "PID" "MSSubClass" "LotFrontage" ...
```

We can subject the data to a PCA and collect the results in summarized form.

```
AmesPCA <- pca(AmesPCAMeths, method = 'svd', center = FALSE, nPcs = 10)
AmesPCA
```

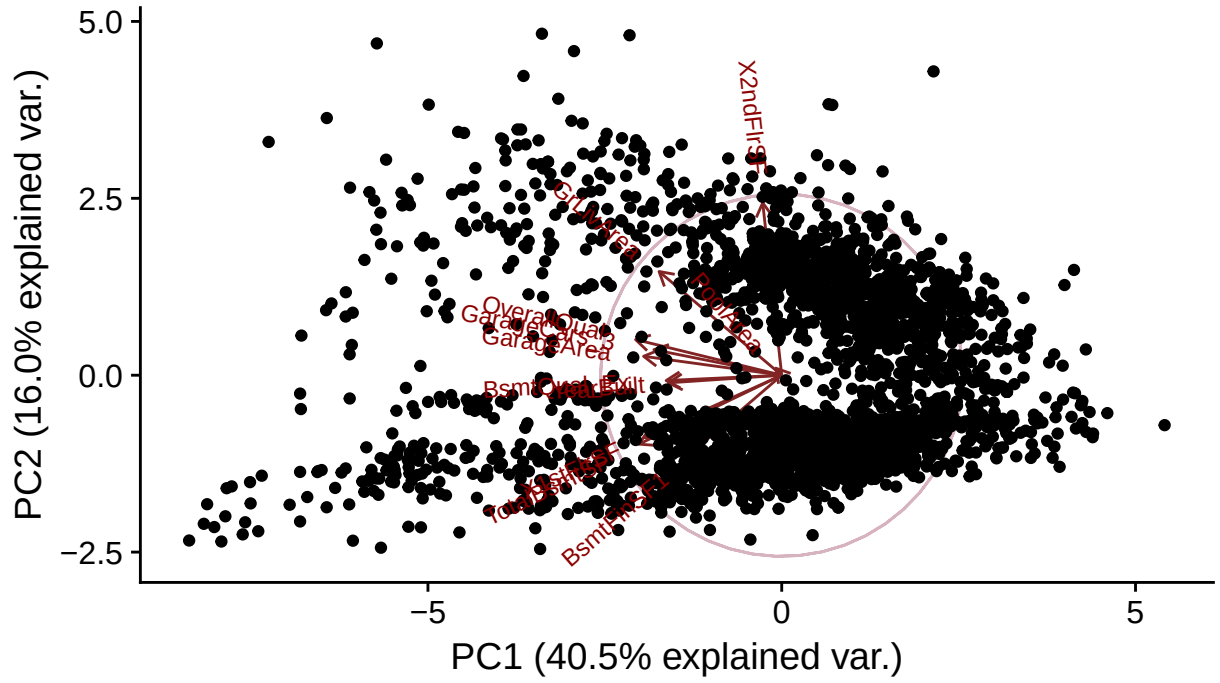
```
## svd calculated PCA
## Importance of component(s):
##                    PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8
## R2             0.4051 0.1599 0.0948 0.07762 0.06793 0.06312 0.05357 0.03212
## Cumulative R2 0.4051 0.5650 0.6598 0.73743 0.80536 0.86848 0.92205 0.95417
##                    PC9    PC10
## R2             0.02657 0.01892
## Cumulative R2 0.98074 0.99966
## 11   Variables
## 2925     Samples
## 0    NAs ( 0 %)
## 10   Calculated component(s)
## Data was NOT mean centered before running PCA
## Data was NOT scaled before running PCA
## Scores structure:
## [1] 2925   10
## Loadings structure:
## [1] 11 10
```

```
summary(AmesPCA)
```

```
## svd calculated PCA
## Importance of component(s):
##                    PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8
## R2              0.4051 0.1599 0.0948 0.07762 0.06793 0.06312 0.05357 0.03212
## Cumulative R2 0.4051 0.5650 0.6598 0.73743 0.80536 0.86848 0.92205 0.95417
##                    PC9    PC10
## R2              0.02657 0.01892
## Cumulative R2 0.98074 0.99966
```

For graphing purposes, we will create a separate object consistent with the requirements of the graphing function. The following chart plots the PC1 values against the PC2 values and generates eigenvectors (the red lines) showing the direction of the variance for the top 10 features.

```
AmesPCA2 <- prcomp(AmesPCAMeths)
ggbiplot::ggbiplot(AmesPCA2, obs.scale = 1, var.scale = 1, ellipse = TRUE, circle = TRUE) +
        scale_color_discrete(name = '') +
        theme(legend.direction = 'horizontal', legend.position = 'top')
```



We are looking for orthogonality between features, and the chart above displays some orthogonal relationships. It appears the nearly vertical red line, *X2ndFlrSF* is orthogonal to *BsmtQual_Ex* and *YearBuilt*, so there are three features we can consider. It also appears the two red lines between 10 o'clock and 11 o'clock, *GrLivArea* and *PoolArea*, are orthogonal to *BsmtFinSF1*. We have another three features to consider. In these two PCs there are for total of six features of interest.

We can also compare the loadings of all 10 PCs to get a sense of their structure and direction. First, we arrange the data set

```
AmesLoad <- loadings(AmesPCA)
AmesLoad
```

```
##                       PC1         PC2         PC3         PC4         PC5
## TotalBsmtSF  -0.37325603 -0.28684388 -0.08907971 -0.23298131  0.12065399
## BsmtFinSF1   -0.23962063 -0.32796687 -0.05961810 -0.10987786  0.25460739
## X2ndFlrSF    -0.04969298  0.71736020 -0.01860356 -0.09896304  0.14318081
## X1stFlrSF    -0.36061095 -0.25941545 -0.23249910 -0.37142447  0.03150384
## PoolArea     -0.01591112  0.02943620 -0.88873267  0.43123913 -0.12590853
## YearBuilt    -0.29634226 -0.03178862  0.22109094  0.60823453  0.44433778
## GrLivArea    -0.32085398  0.43172401 -0.20485842 -0.38736125  0.14306654
## OverallQual  -0.38225826  0.14832171  0.09614477  0.18885229  0.19689054
## GarageCars_3 -0.33434865  0.10624966  0.13757020  0.07275359 -0.54387139
## BsmtQual_Ex  -0.30228738 -0.02218145  0.15048432  0.11740167 -0.57234865
## GarageArea   -0.36117171  0.07882471  0.05098734  0.15871407 -0.05676664
##                       PC6          PC7         PC8         PC9        PC10
## TotalBsmtSF  -0.132950791  0.139141470 -0.21480213 -0.05679882  0.78436251
## BsmtFinSF1    0.734691859 -0.434577399 -0.07911976 -0.03701913 -0.13526131
## X2ndFlrSF     0.249117506 -0.057188776 -0.04011998  0.07732950  0.23150227
## X1stFlrSF    -0.272545590  0.168659245  0.12576243  0.21091715 -0.43502232
## PoolArea      0.046999164  0.003965257 -0.05251243 -0.02359741  0.04089156
## YearBuilt    -0.063383381  0.125706481 -0.03911778  0.52251356 -0.02112934
## GrLivArea     0.004748772  0.085152128  0.06309297  0.22801840 -0.13491878
## OverallQual  -0.030143513  0.288562205 -0.22856330 -0.73656636 -0.26316532
## GarageCars_3 -0.154117849 -0.385718250 -0.58883806  0.16213706 -0.10657062
## BsmtQual_Ex   0.450513720  0.467834658  0.33048142  0.08359789  0.07797027
## GarageArea   -0.268958827 -0.534865337  0.64320791 -0.20871247  0.12894888
```

It doesn't look as though there are many strong correlations between our 10 PCs and the 11 features composing them. PC2 appears correlated to *X2ndFlrSF* and PC6 seems to be correlated to *BsmtFinSF1*. PC3 appears corrlated to *PoolArea*. PC9 seems correlated to *OverallQual* and PC10 to *TotalBsmtSF*, but that's it.

Then we structure it for plotting.

```
AmesMelt <- melt(AmesLoad)
colnames(AmesMelt)[1] <- 'names'
AmesMelt <- arrange(AmesMelt, names)
AmesMelt$X2 <- gsub('PC1', 'PC01', AmesMelt$X2)
AmesMelt$X2 <- gsub('PC2', 'PC02', AmesMelt$X2)
AmesMelt$X2 <- gsub('PC3', 'PC03', AmesMelt$X2)
AmesMelt$X2 <- gsub('PC4', 'PC04', AmesMelt$X2)
AmesMelt$X2 <- gsub('PC5', 'PC05', AmesMelt$X2)
AmesMelt$X2 <- gsub('PC6', 'PC06', AmesMelt$X2)
AmesMelt$X2 <- gsub('PC7', 'PC07', AmesMelt$X2)
AmesMelt$X2 <- gsub('PC8', 'PC08', AmesMelt$X2)
AmesMelt$X2 <- gsub('PC9', 'PC09', AmesMelt$X2)
AmesMelt$X2 <- gsub('PC010', 'PC10', AmesMelt$X2)
```

Run a sanity check to make sure the data are in the proper structure.

```
AmesMelt
```

```
##           names   X2        value
## 1     BsmtFinSF1 PC01 -0.239620634
## 2     BsmtFinSF1 PC02 -0.327966870
## 3     BsmtFinSF1 PC03 -0.059618101
## 4     BsmtFinSF1 PC04 -0.109877858
```

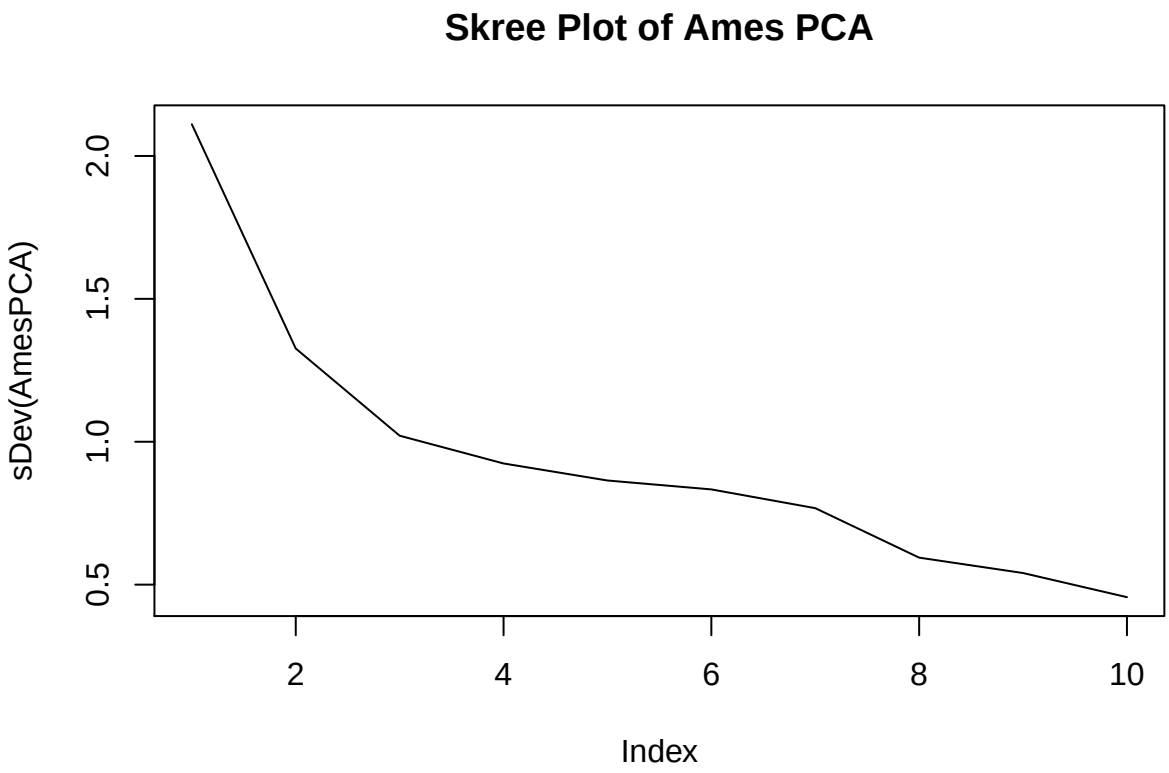
```
## 5     BsmtFinSF1 PC05  0.254607389
## 6     BsmtFinSF1 PC06  0.734691859
## 7     BsmtFinSF1 PC07 -0.434577399
## 8     BsmtFinSF1 PC08 -0.079119756
## 9     BsmtFinSF1 PC09 -0.037019132
## 10    BsmtFinSF1 PC10 -0.135261313
## 11   BsmtQual_Ex PC01 -0.302287381
## 12   BsmtQual_Ex PC02 -0.022181455
## 13   BsmtQual_Ex PC03  0.150484321
## 14   BsmtQual_Ex PC04  0.117401671
## 15   BsmtQual_Ex PC05 -0.572348646
## 16   BsmtQual_Ex PC06  0.450513720
## 17   BsmtQual_Ex PC07  0.467834658
## 18   BsmtQual_Ex PC08  0.330481419
## 19   BsmtQual_Ex PC09  0.083597893
## 20   BsmtQual_Ex PC10  0.077970268
## 21    GarageArea PC01 -0.361171708
## 22    GarageArea PC02  0.078824709
## 23    GarageArea PC03  0.050987338
## 24    GarageArea PC04  0.158714066
## 25    GarageArea PC05 -0.056766644
## 26    GarageArea PC06 -0.268958827
## 27    GarageArea PC07 -0.534865337
## 28    GarageArea PC08  0.643207912
## 29    GarageArea PC09 -0.208712469
## 30    GarageArea PC10  0.128948881
## 31  GarageCars_3 PC01 -0.334348654
## 32  GarageCars_3 PC02  0.106249656
## 33  GarageCars_3 PC03  0.137570200
## 34  GarageCars_3 PC04  0.072753592
## 35  GarageCars_3 PC05 -0.543871388
## 36  GarageCars_3 PC06 -0.154117849
## 37  GarageCars_3 PC07 -0.385718250
## 38  GarageCars_3 PC08 -0.588838057
## 39  GarageCars_3 PC09  0.162137058
## 40  GarageCars_3 PC10 -0.106570625
## 41     GrLivArea PC01 -0.320853979
## 42     GrLivArea PC02  0.431724007
## 43     GrLivArea PC03 -0.204858424
## 44     GrLivArea PC04 -0.387361250
## 45     GrLivArea PC05  0.143066542
## 46     GrLivArea PC06  0.004748772
## 47     GrLivArea PC07  0.085152128
## 48     GrLivArea PC08  0.063092969
## 49     GrLivArea PC09  0.228018403
## 50     GrLivArea PC10 -0.134918776
## 51   OverallQual PC01 -0.382258261
## 52   OverallQual PC02  0.148321709
## 53   OverallQual PC03  0.096144769
## 54   OverallQual PC04  0.188852289
## 55   OverallQual PC05  0.196890541
## 56   OverallQual PC06 -0.030143513
## 57   OverallQual PC07  0.288562205
## 58   OverallQual PC08 -0.228563300
```

```
## 59   OverallQual PC09 -0.736566359
## 60   OverallQual PC10 -0.263165324
## 61      PoolArea PC01 -0.015911121
## 62      PoolArea PC02  0.029436195
## 63      PoolArea PC03 -0.888732675
## 64      PoolArea PC04  0.431239127
## 65      PoolArea PC05 -0.125908526
## 66      PoolArea PC06  0.046999164
## 67      PoolArea PC07  0.003965257
## 68      PoolArea PC08 -0.052512432
## 69      PoolArea PC09 -0.023597411
## 70      PoolArea PC10  0.040891558
## 71   TotalBsmtSF PC01 -0.373256030
## 72   TotalBsmtSF PC02 -0.286843882
## 73   TotalBsmtSF PC03 -0.089079710
## 74   TotalBsmtSF PC04 -0.232981314
## 75   TotalBsmtSF PC05  0.120653988
## 76   TotalBsmtSF PC06 -0.132950791
## 77   TotalBsmtSF PC07  0.139141470
## 78   TotalBsmtSF PC08 -0.214802133
## 79   TotalBsmtSF PC09 -0.056798818
## 80   TotalBsmtSF PC10  0.784362511
## 81     X1stFlrSF PC01 -0.360610950
## 82     X1stFlrSF PC02 -0.259415448
## 83     X1stFlrSF PC03 -0.232499103
## 84     X1stFlrSF PC04 -0.371424469
## 85     X1stFlrSF PC05  0.031503839
## 86     X1stFlrSF PC06 -0.272545590
## 87     X1stFlrSF PC07  0.168659245
## 88     X1stFlrSF PC08  0.125762434
## 89     X1stFlrSF PC09  0.210917152
## 90     X1stFlrSF PC10 -0.435022319
## 91     X2ndFlrSF PC01 -0.049692977
## 92     X2ndFlrSF PC02  0.717360196
## 93     X2ndFlrSF PC03 -0.018603556
## 94     X2ndFlrSF PC04 -0.098963043
## 95     X2ndFlrSF PC05  0.143180813
## 96     X2ndFlrSF PC06  0.249117506
## 97     X2ndFlrSF PC07 -0.057188776
## 98     X2ndFlrSF PC08 -0.040119983
## 99     X2ndFlrSF PC09  0.077329501
## 100    X2ndFlrSF PC10  0.231502274
## 101    YearBuilt PC01 -0.296342260
## 102    YearBuilt PC02 -0.031788619
## 103    YearBuilt PC03  0.221090944
## 104    YearBuilt PC04  0.608234528
## 105    YearBuilt PC05  0.444337785
## 106    YearBuilt PC06 -0.063383381
## 107    YearBuilt PC07  0.125706481
## 108    YearBuilt PC08 -0.039117780
## 109    YearBuilt PC09  0.522513562
## 110    YearBuilt PC10 -0.021129343
```

We will come back to this in the next section on clustering.

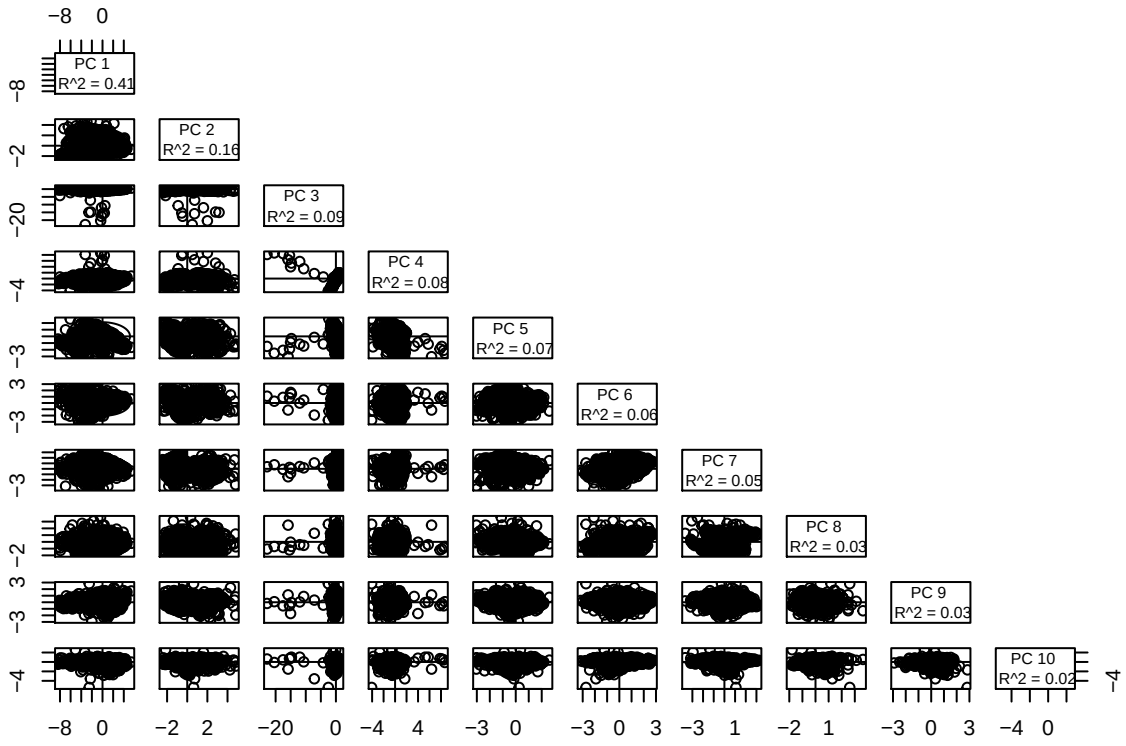
A quick way to determine which features are most important is to plot the standard deviations from the PCA for each principal component and see "where the line breaks." The following graph suggests two features definitely, though an argument can be made that three, possibly four, features contribute more to our understanding of the data. Even if we choose to use just two PCs, we know from the chart above there are six features arranged in an orthogonal manner.

```
plot(sDev(AmesPCA), type = "lines", main = 'Skree Plot of Ames PCA')
```



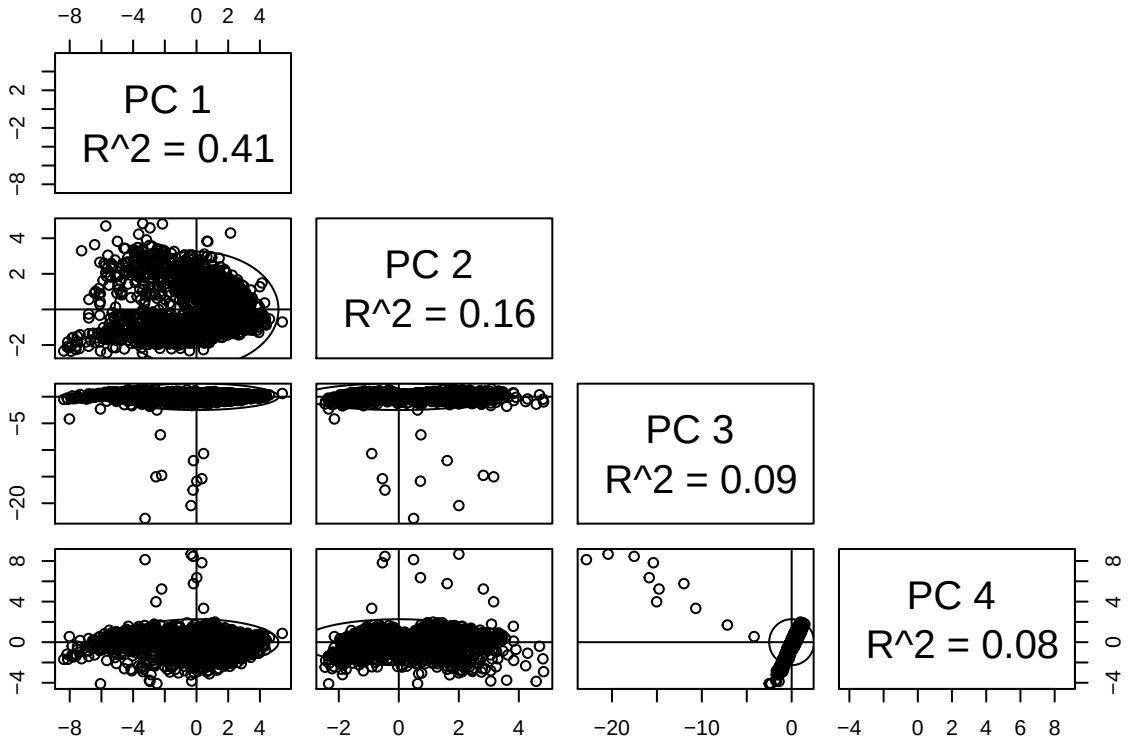
Another way to look at the PCA results is with a pair plot. But taking all 10 PCs into consideration might be a bit much.

```
plotPcs(AmesPCA)
```



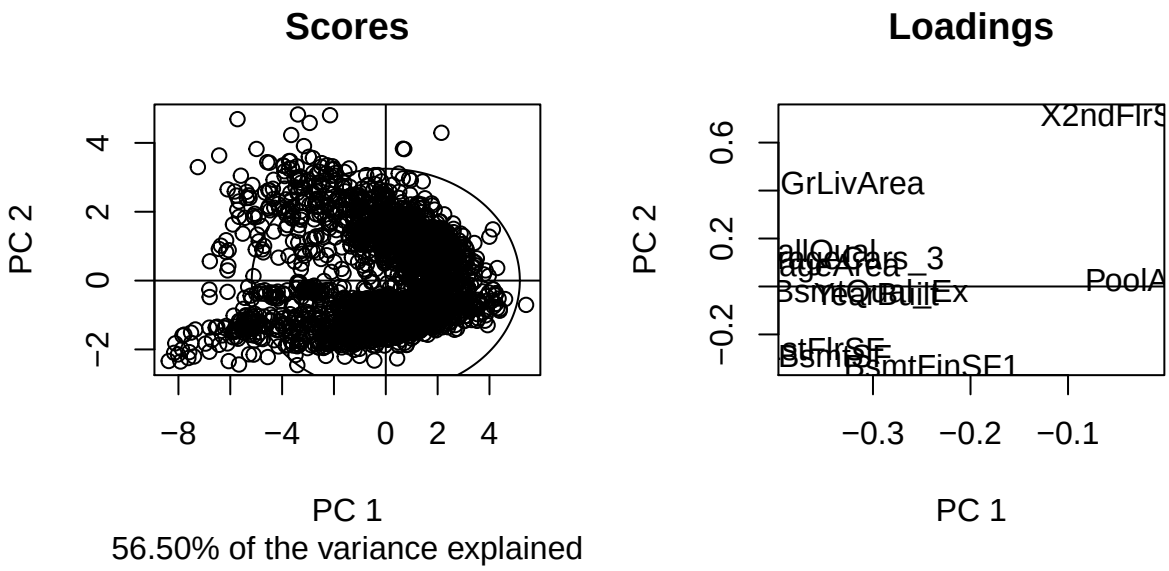
Definitely. Since our skree plot suggests three or four PCs are sufficient, let's pare down the pairs plots to two pair.

```
plotPcs(AmesPCA, pcs = 1:4)
```

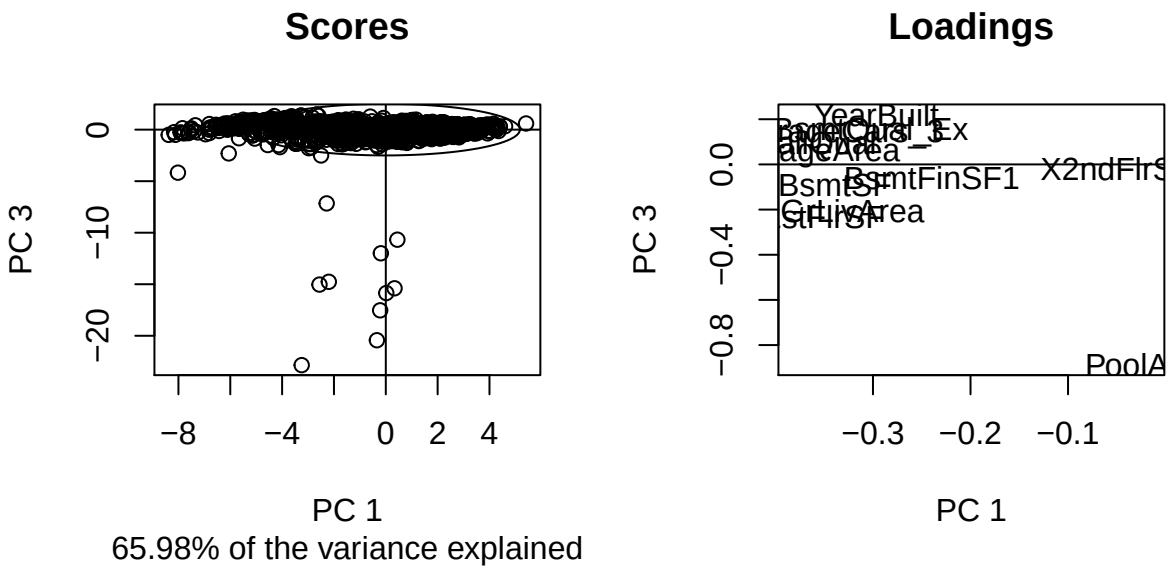


The following set of graphs compares PC1 to PC2, PC3, and PC4 in greater detail and includes analysis of the loadings.

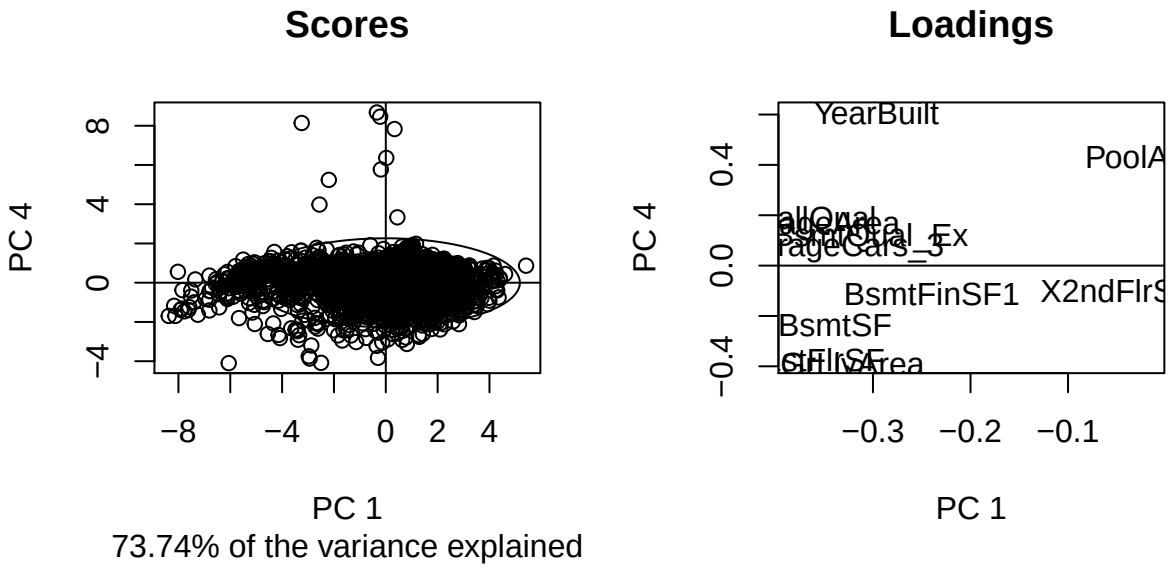
```
slplot(AmesPCA, pcs = c(1, 2))
```




```
slplot(AmesPCA, pcs = c(1, 3))
```

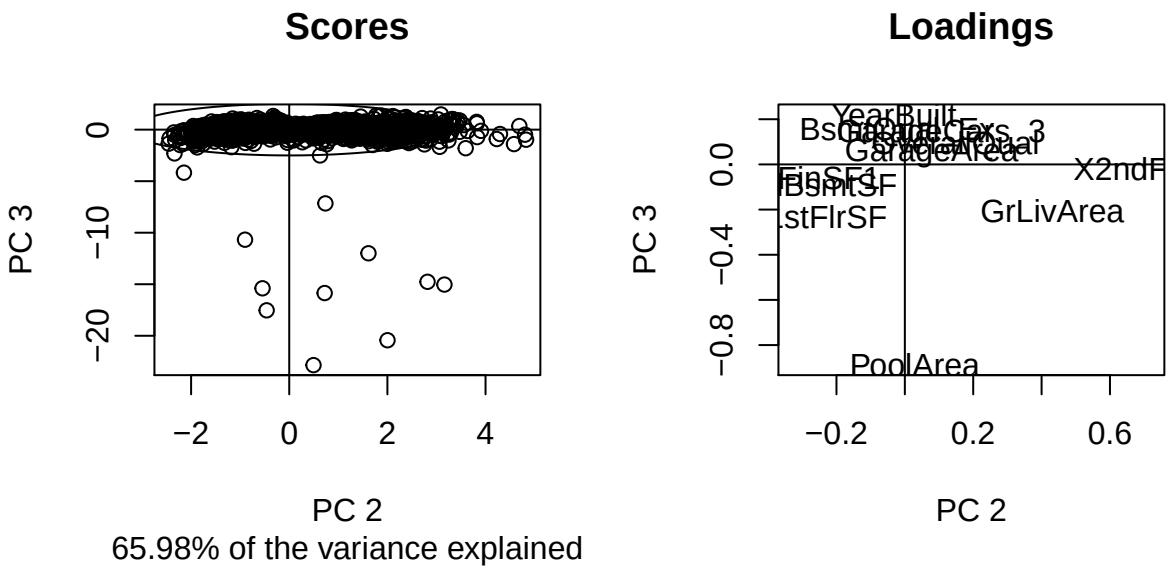



```
slplot(AmesPCA, pcs = c(1, 4))
```

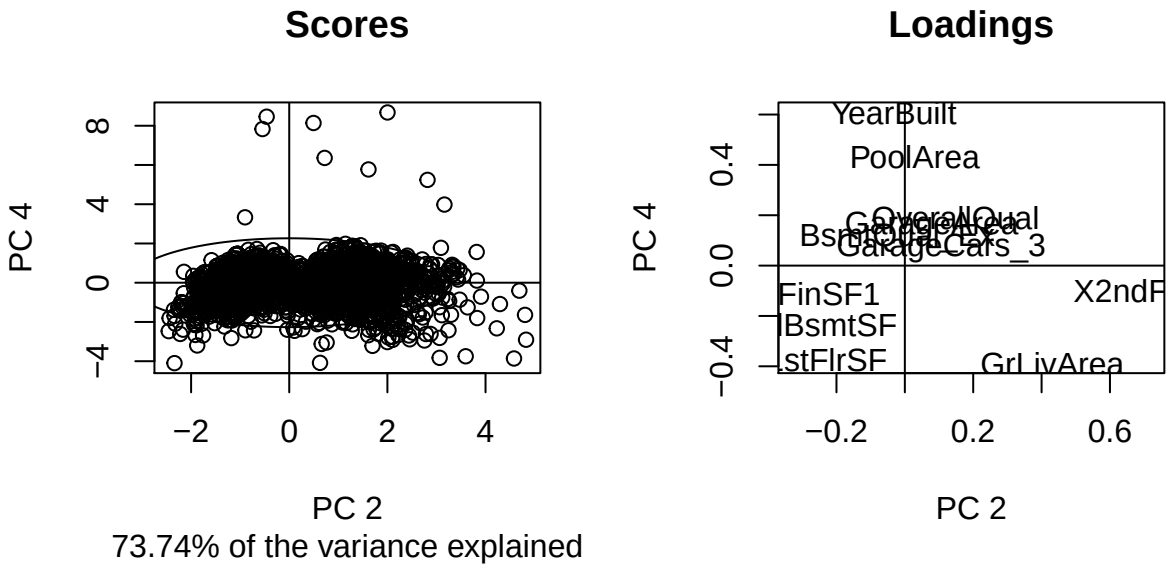


Here, we compare PC2 with PC3 and PC4.

```
slplot(AmesPCA, pcs = c(2, 3))
```

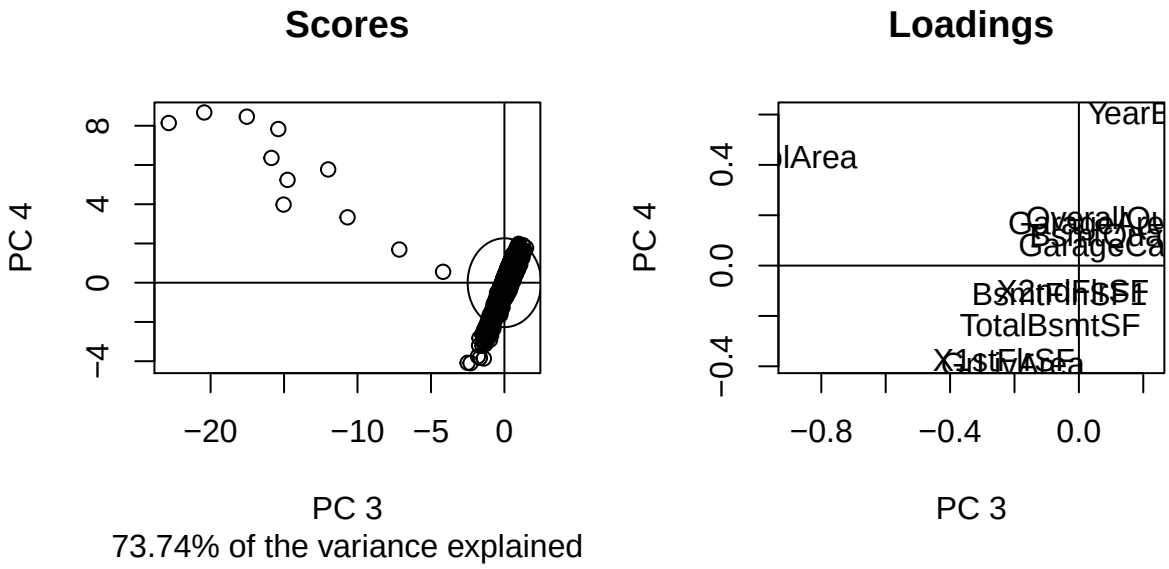



```
slplot(AmesPCA, pcs = c(2, 4))
```



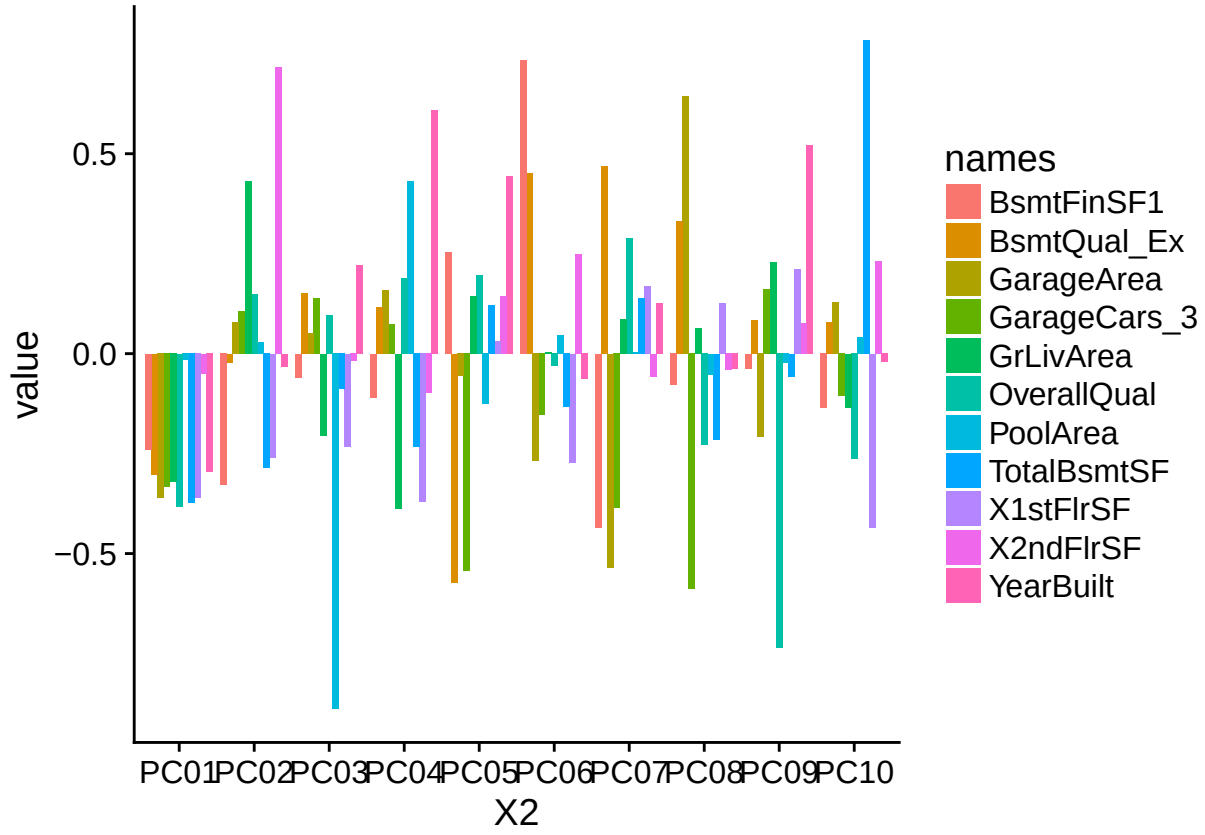
We complete the set with a graph of PC3 and PC4.

```
slplot(AmesPCA, pcs = c(3, 4))
```



Then produce a visualization of the loadings.

```
ggplot(AmesMelt, aes(x = X2, y = value, fill = names)) + geom_bar(stat = 'identity',
        position = 'dodge') + guides(fill = guide_legend(reverse = FALSE))
```



As mentioned earlier, there are five features that dominate the PCs. It doesn't look as though there are many strong correlations between our 10 PCs and the 11 features composing them. In PC2 there is *X2ndFlrSF*. PC6 is dominated by *BsmtFinSF1* while *PoolArea* dominates PC3. PC9's dominant feature is *OverallQual* and PC10 is characterized by *TotalBsmtSF*. We can see these features in the loading graph above. This suggest we should include these five when building our multilinear regression model.

We also mentioned earlier that PC1 consists of negative values only, suggesting the most important PC, the one that accounts for most of the variation in the data, is related somehow to below average housing features.

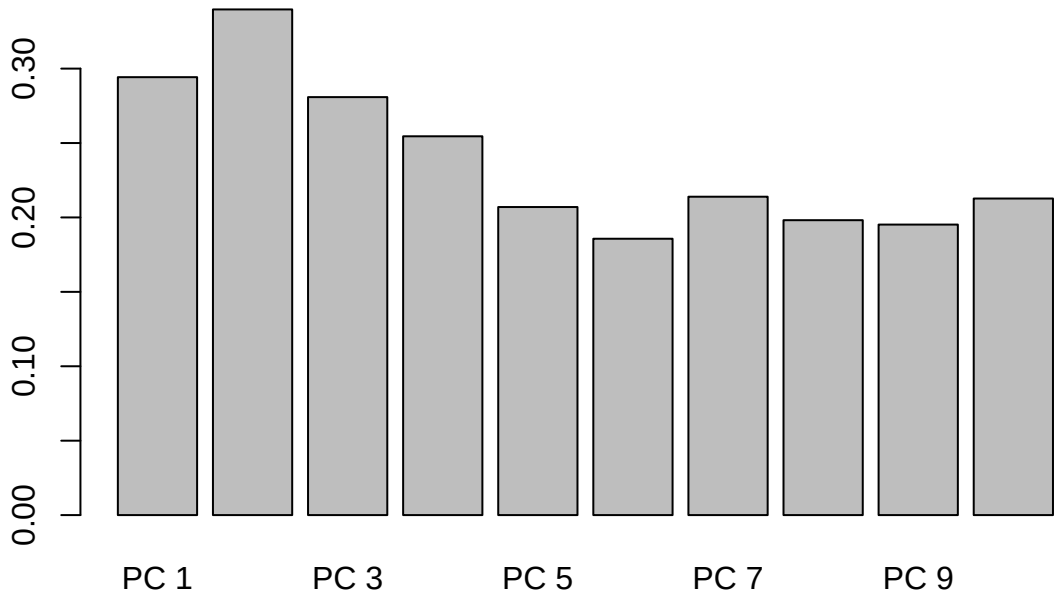
It is possible to cross validate the PCA results by using the *Q2* measure, a cross-validation version of *R2*. It is the ratio of variance predicted independently by the PCA model.

```
q2svd <- Q2(AmesPCA, AmesPCAMeths, fold = 15, nruncv = 1)
q2svd
```

```
##      PC 1      PC 2      PC 3      PC 4      PC 5      PC 6      PC 7
## 0.2942543 0.3397652 0.2808441 0.2545190 0.2069783 0.1856758 0.2139185
##      PC 8      PC 9     PC 10
## 0.1981492 0.1951924 0.2127118
```

This is a "leave one out" cross validation method. The results suggest that three or four PCs are valuable to estimating the data structure. The following plot shows values dropping off after PC4.

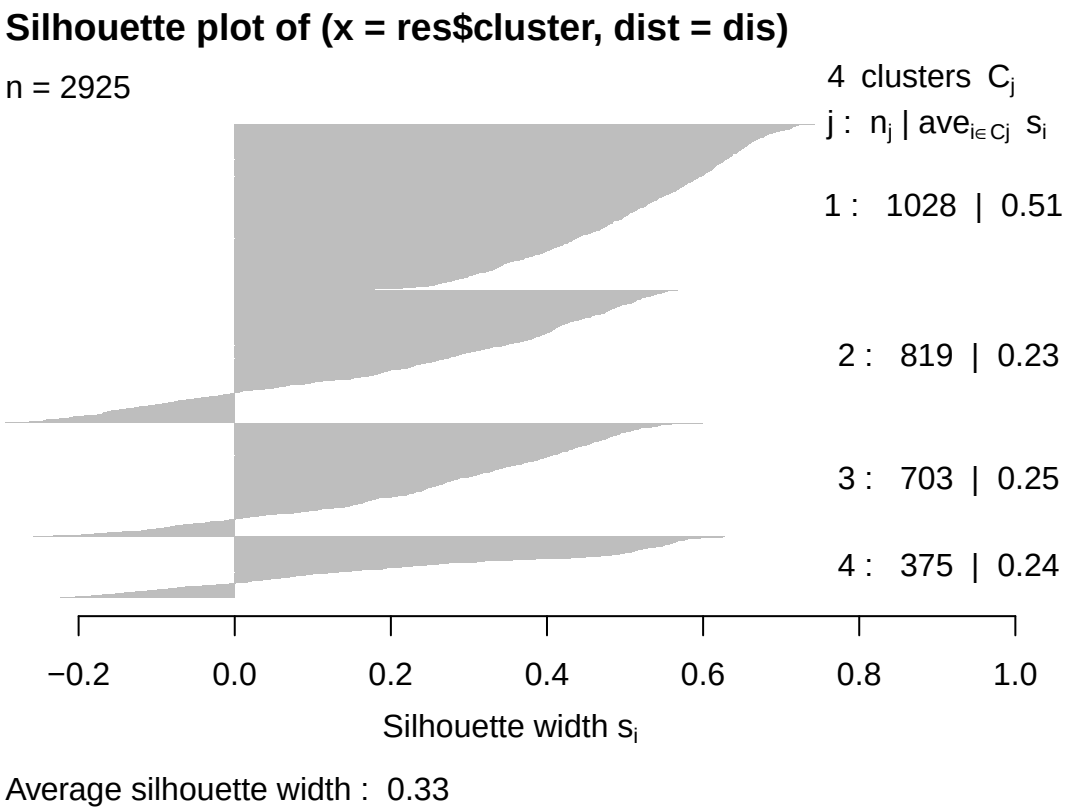
```
barplot(q2svd)
```



Clustering

We have an idea of how many PCs are important to our understanding of the *Ames* data structure, so we can use clustering to confirm our findings. We will use a k-means cluster method so it is important to measure clusters in terms appropriate for k-means analysis. In this case, we will work with squared distances. The resulting silhouette chart demonstrates the results.

```
dis <- dist(AmesPCAMeths)^2
res <- kmeans(AmesPCAMeths, 4)
sil <- silhouette(res$cluster, dis)
windows()
plot(sil)
```



The silhouette chart indicates the four PCs tend to cluster well together, especially PC1 in which all cluster in the same direction. In PC2, PC3, and PC4 most clustering is in the positive direction.

We can consider the four clusters from a different perspective with scatterplots. First we calculate the data and determine which cluster each data point belongs to.

```
AmesCl <- kmeans(AmesPCAMeths, 4, nstart = 10)
AmesCl
```

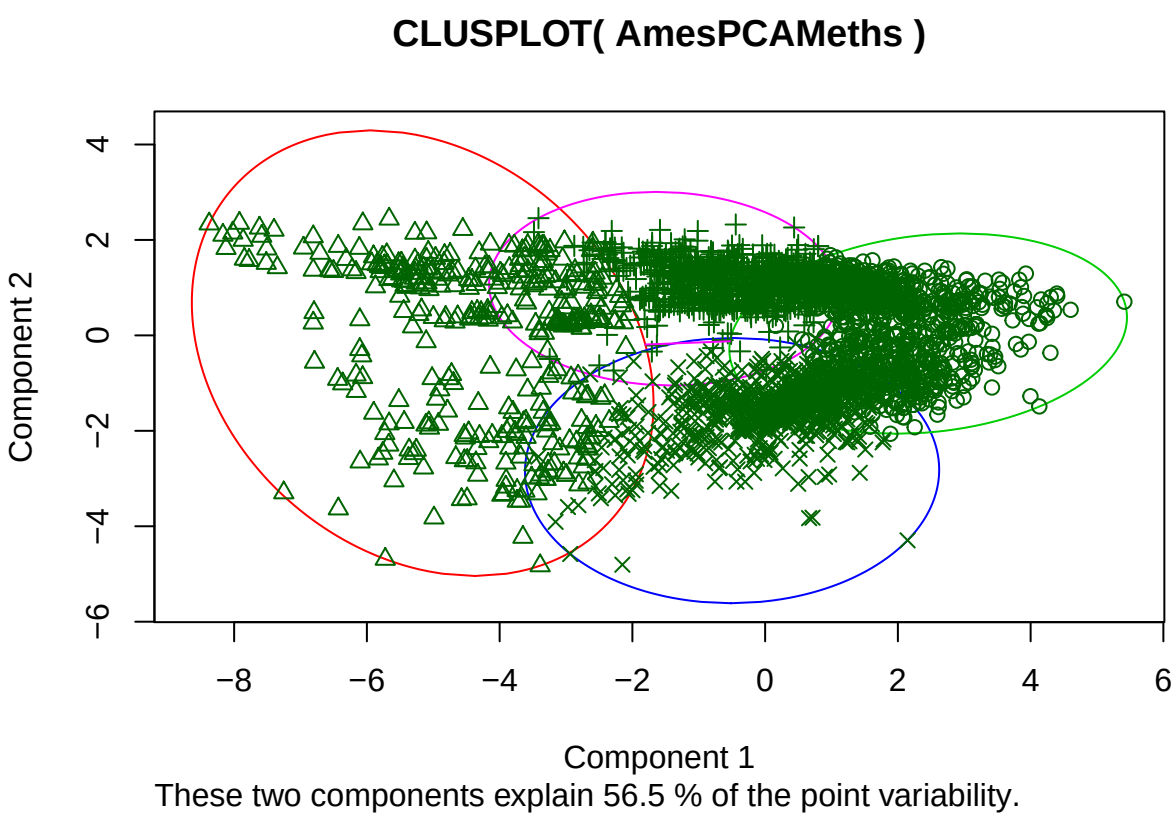
```
## K-means clustering with 4 clusters of sizes 1028, 375, 819, 703
##
## Cluster means:
##   TotalBsmtSF BsmtFinSF1  X2ndFlrSF  X1stFlrSF    PoolArea  YearBuilt
## 1  -0.6694657 -0.3847359 -0.2996653 -0.6469502 -0.05743592 -0.8109677
## 2   1.3842824  0.8425579  0.1513656  1.3188245 -0.04578695  1.0051568
## 3   0.5841330  0.3838242 -0.7693100  0.6130855  0.06038876  0.3227847
## 4  -0.4399717 -0.3340011  1.2537108 -0.4717090  0.03805951  0.2736562
##    GrLivArea OverallQual GarageCars_3 BsmtQual_Ex GarageArea
## 1 -0.7586944 -0.82488237   -0.3599980  -0.2972776 -0.7646579
## 2  1.1474140  1.50144830    2.1220468   1.7790641  1.4552508
## 3 -0.2055015  0.07028607   -0.3364084  -0.2251408  0.1857148
## 4  0.7367899  0.32343055   -0.2136147  -0.2520019  0.1255318
##
## Clustering vector:
##    [1] 3 1 3 3 4 4 3 3 3 4 4 3 4 3 3 2 4 2 1 3 3 3 4 3 3 1 1 1 3 1 1 1 3 4
##   [35] 1 4 2 2 2 2 3 2 2 3 2 3 2 2 2 3 4 3 3 3 4 4 4 4 4 2 2 3 2 2 4 2 2 3
##   [69] 4 3 4 3 4 4 4 1 4 1 1 4 3 4 1 4 3 1 1 1 1 3 4 2 2 2 4 4 4 4 4 3 4 4
##  [103] 4 4 2 4 3 3 4 3 3 4 3 4 4 4 3 4 3 3 1 3 3 4 3 3 3 1 3 4 1 1 1 3 1 3 4
```

```
##  [137] 3 3 3 3 3 3 3 1 3 3 3 3 1 1 1 1 1 1 1 1 1 3 3 4 1 1 1 1 1 3 3 4 1
##  [171] 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 4 4 3 1 4 1 1 4 1 1 1 1 1 1 1 1 4 4 1
##  [205] 1 1 1 1 4 3 1 1 1 1 1 1 3 3 3 3 1 1 1 1 1 3 4 4 2 3 1 1 3 4 1 1 1 3
##  [239] 1 3 3 3 3 1 4 3 4 4 3 2 3 3 3 2 1 3 3 4 4 1 4 1 1 2 3 4 4 3 3 3 3 3
##  [273] 3 1 1 1 3 1 3 3 1 1 1 1 3 1 1 1 1 1 4 1 4 4 4 4 3 4 3 3 3 1 1 1 1 1
##  [307] 1 1 2 2 3 2 3 3 3 3 3 4 2 4 4 2 3 3 4 3 1 1 1 1 1 1 1 1 3 1 4 4 1 3
##  [341] 4 1 3 2 4 4 4 4 2 4 2 3 3 3 4 4 4 4 4 4 4 3 4 4 3 4 3 3 2 2 2 4 4 3 3 3
##  [375] 4 3 4 3 3 2 3 3 4 4 4 4 4 4 3 3 3 3 4 4 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [409] 3 4 4 3 1 4 4 4 4 4 1 1 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [443] 2 2 2 3 3 2 2 4 2 3 3 3 3 3 2 2 2 2 2 3 3 4 4 4 4 4 2 2 3 3 3 3 4 4
##  [477] 4 4 4 4 4 4 4 3 4 2 4 4 4 4 4 4 2 4 4 4 2 2 2 2 2 2 2 2 4 4 2 4 2 2 2 2
##  [511] 2 2 4 2 4 4 3 3 3 3 2 2 3 2 2 2 3 3 2 4 3 3 4 3 3 1 4 2 3 3 1 3 2 4
##  [545] 3 3 4 1 3 4 4 4 1 1 1 1 1 1 1 1 1 1 2 4 4 4 4 3 3 4 4 4 4 3 3 3 4
##  [579] 3 4 3 3 4 4 4 4 4 3 3 3 4 3 2 4 1 3 1 4 3 3 1 1 1 3 3 2 3 3 3 3 4 3 1
##  [613] 3 1 1 1 3 3 3 1 1 1 3 3 3 3 4 4 3 1 1 3 3 1 3 3 1 3 1 3 3 3 1 1 1 1
##  [647] 1 1 1 1 1 4 1 1 1 1 1 1 1 1 1 4 1 1 1 1 4 3 3 4 1 1 3 3 1 1 1 4 1 4
##  [681] 1 1 1 1 3 1 1 4 3 1 1 1 1 1 4 1 1 1 1 4 1 4 1 1 1 1 1 1 1 1 1 1 1 1
##  [715] 1 4 4 4 1 4 1 1 1 1 1 4 1 1 3 1 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 4
##  [749] 4 4 4 4 4 4 4 1 1 1 1 1 1 1 3 3 3 1 1 1 1 3 4 3 1 1 1 1 1 4 3 3 1 1
##  [783] 3 1 1 3 1 1 3 1 1 4 1 1 1 1 1 1 1 3 3 3 2 4 4 4 1 1 3 4 1 1 4 3 1 2
##  [817] 2 2 2 2 2 2 2 4 2 2 3 3 2 3 2 4 3 4 4 3 3 4 4 3 3 3 4 3 3 4 4 3 4 3
##  [851] 3 3 1 1 1 3 1 3 1 1 1 1 2 4 3 3 2 4 4 3 3 2 3 4 4 3 3 3 3 2 4 3 4 4
##  [885] 1 3 1 4 4 1 3 2 3 1 1 3 1 1 1 3 3 1 1 1 4 1 1 1 1 4 1 4 1 1 1 4 1 1
##  [919] 4 1 1 1 3 1 1 4 1 4 3 1 1 3 1 4 4 4 4 3 3 1 4 1 1 1 2 1 4 2 2 4 1 3
##  [953] 4 3 3 3 3 2 4 2 2 2 3 4 3 3 4 2 2 1 1 3 3 1 1 1 3 1 1 1 1 1 3 1 3 4
##  [987] 4 3 4 3 3 4 4 4 4 4 4 4 3 4 2 2 3 3 4 4 4 4 4 4 3 3 2 4 2 3 2 3 4 3 3 3
## [1021] 3 3 4 3 3 4 1 4 3 3 4 3 2 4 3 1 1 1 3 1 1 1 1 1 1 1 3 4 3 1 2 2 2 2
## [1055] 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 3 2 2 2 3 3 4 4 3 2 2 3 3 3 4 4 3
## [1089] 4 4 4 3 4 4 4 4 4 4 4 4 2 2 2 2 2 4 2 2 2 4 4 3 2 4 2 2 3 2 3 2 2 2 3 3
## [1123] 3 4 4 4 2 4 2 3 4 3 4 4 1 3 3 4 3 4 4 3 4 4 4 3 4 3 3 1 1 1 1 1 1 3
## [1157] 4 4 2 3 4 3 4 4 4 4 4 4 3 2 2 3 2 4 4 4 4 4 4 4 3 3 3 3 2 3 4 3 3 3 3 3
## [1191] 3 4 4 1 1 3 3 3 4 4 4 4 3 4 1 1 1 3 3 3 3 3 3 3 1 1 3 1 1 1 1 1 1 3
## [1225] 1 1 3 1 4 3 3 3 3 4 1 1 3 1 3 3 3 3 3 1 1 1 3 1 3 3 1 1 1 3 1 1 1 1
## [1259] 3 1 4 1 3 3 3 3 1 1 1 3 3 1 3 1 1 1 1 3 1 1 4 4 1 1 1 1 1 1 4 1 1 1
## [1293] 4 1 1 1 4 1 1 1 1 4 1 1 1 1 4 1 1 1 1 1 4 4 1 1 4 1 4 1 4 1 4 1 4 1
## [1327] 1 1 4 1 1 1 1 1 1 1 1 4 1 1 1 1 1 1 1 4 4 4 1 1 1 4 4 4 1 1 4 4 1 1
## [1361] 4 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 1 1 1 1 1 3 3 3 1 1 1 3 3 3 1
## [1395] 1 3 1 3 3 4 3 3 3 3 1 3 4 3 1 4 3 1 1 1 1 1 1 1 4 4 3 4 4 4 4 2 2 2 4
## [1429] 2 3 4 3 1 3 3 4 4 4 4 4 3 3 4 4 4 1 1 3 3 3 3 1 1 1 3 4 2 4 4 4 4 4 2 2
## [1463] 4 4 4 4 4 4 4 3 3 3 2 3 4 3 3 4 3 2 3 4 3 3 3 3 1 1 1 3 1 3 1 3 4 3 4
## [1497] 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 3 1 1 4 4 1 4 1 1 1 4 1 4
## [1531] 1 1 1 1 1 4 2 4 4 3 3 1 4 1 1 1 1 1 1 1 1 4 1 1 1 1 1 1 1 2 2 2 2 2 2
## [1565] 4 3 3 4 3 3 3 2 2 2 4 4 4 4 3 3 3 3 2 2 2 3 2 3 3 1 3 3 3 1 1 1 1 3
## [1599] 1 1 1 1 1 1 3 1 4 1 4 1 3 3 3 3 4 4 4 4 4 4 3 4 4 4 3 4 3 4 4 4 4 3 3 4
## [1633] 3 3 2 2 2 3 2 3 2 2 3 3 4 4 3 4 4 4 3 3 4 3 4 3 3 4 4 3 4 1 3 4 3 1
## [1667] 1 1 1 1 1 1 3 3 1 1 1 1 1 1 4 3 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1701] 2 4 2 2 2 2 2 2 2 2 4 2 2 3 2 3 3 3 3 3 3 3 2 2 2 3 3 4 4 4 4 4 4 3 4
## [1735] 3 3 4 4 4 3 2 2 2 3 2 3 4 4 3 4 4 4 4 4 4 4 4 4 4 2 4 2 2 2 2 4 4 2 2
## [1769] 2 2 4 2 3 2 2 2 3 2 2 2 3 2 2 2 3 3 3 2 4 2 3 3 4 4 2 4 2 2 4 3 2 4
## [1803] 4 1 3 3 3 4 4 3 3 3 1 1 3 1 1 1 1 3 3 3 3 1 4 4 3 3 3 2 2 3 3 3 3 3
## [1837] 3 3 4 4 4 4 4 4 4 4 4 4 3 2 2 3 3 3 3 3 4 3 2 4 4 1 4 3 3 3 3 4 3 3 3
## [1871] 1 1 3 3 3 3 1 1 1 1 1 1 1 1 1 3 1 3 3 3 4 3 3 3 3 1 1 1 1 1 1 1 4 1
## [1905] 4 1 1 3 1 3 1 1 4 1 1 3 1 3 1 3 1 3 1 1 3 3 1 1 3 1 1 1 1 1 1 1 1 3
## [1939] 1 1 1 3 4 1 1 1 1 1 1 4 1 1 4 1 1 4 1 1 1 1 1 1 3 3 3 1 4 4 1 1 1 1
```

```
## [1973] 1 1 3 1 1 1 1 1 1 1 1 1 1 1 4 1 1 1 4 1 4 1 4 4 1 3 4 1 1 1 1 1 1 4
## [2007] 1 1 1 1 1 1 1 1 1 1 1 1 4 4 1 1 1 1 4 1 1 4 1 4 1 1 4 4 1 1 1 1 4 1
## [2041] 1 4 4 1 1 1 3 1 1 1 1 1 1 1 1 1 1 1 1 3 1 3 4 1 1 3 1 1 3 3 4 1 3 1
## [2075] 3 3 4 1 1 3 1 1 3 1 1 1 4 3 4 4 2 4 4 2 2 2 2 2 3 2 2 3 4 3 3 2 3 3
## [2109] 4 3 4 4 4 4 4 3 3 1 3 4 4 1 3 1 4 4 4 4 4 4 3 4 1 1 3 3 1 1 1 1 1 1 2
## [2143] 4 3 3 4 4 4 3 4 4 2 4 3 3 4 3 4 4 3 3 3 4 4 3 1 1 1 1 3 1 3 3 4 3 3
## [2177] 4 3 1 1 1 1 1 1 1 1 1 1 1 4 4 1 1 1 1 1 1 1 1 1 4 1 1 1 4 4 4 1 2 4
## [2211] 1 4 1 4 4 3 4 3 3 3 1 3 3 3 3 3 1 1 1 1 1 4 1 1 1 1 1 1 3 1 3 3 3 3
## [2245] 3 3 3 3 1 3 3 3 3 3 3 3 3 3 2 2 3 3 4 3 3 2 3 2 4 2 2 2 4 1 3 1 1 3 3
## [2279] 3 3 1 1 1 1 1 1 3 3 3 3 4 1 1 3 1 3 1 1 1 1 3 1 2 3 3 4 4 4 4 4 4 4 4
## [2313] 2 4 4 2 4 2 3 4 4 3 4 3 2 2 2 2 2 2 2 2 4 3 2 2 2 3 2 4 4 3 3 4 3 4
## [2347] 4 3 4 4 3 4 3 3 1 1 1 1 3 1 1 1 1 1 1 1 1 3 1 1 3 4 2 2 2 2 2 2 2 2
## [2381] 2 2 2 4 4 2 2 4 2 2 2 4 2 2 2 2 4 2 3 2 3 3 2 2 3 3 4 4 4 4 3 4 4 4
## [2415] 3 3 3 3 3 3 3 3 3 2 2 4 4 4 4 4 4 4 4 4 4 4 4 2 2 2 2 2 4 4 2 2 2 2
## [2449] 2 4 2 2 2 2 3 3 2 3 2 4 4 4 3 3 3 1 3 1 3 3 4 4 4 4 4 4 3 3 4 1 1 1 3
## [2483] 4 3 1 1 1 1 1 1 1 1 1 3 4 2 2 4 2 3 3 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3
## [2517] 3 2 3 4 3 4 4 4 3 3 1 3 4 3 1 1 1 3 3 4 1 1 3 3 3 3 4 1 3 3 1 1 1 1
## [2551] 1 1 1 3 3 1 3 1 1 1 3 3 3 4 1 4 1 3 3 3 3 3 3 1 3 1 3 1 3 1 1 1 1 1
## [2585] 1 2 3 2 1 1 1 1 1 1 1 3 4 1 1 4 1 3 1 3 4 3 1 3 1 1 3 1 3 3 1 1 1 1
## [2619] 1 1 1 1 3 1 1 4 1 3 1 1 4 1 1 1 1 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [2653] 4 1 4 4 1 4 1 4 1 2 1 1 1 1 1 4 4 1 1 1 1 1 1 1 1 1 1 1 4 1 1 1 1
## [2687] 1 1 1 1 1 1 1 1 3 1 1 1 1 3 3 1 1 3 3 1 1 1 4 3 3 3 3 3 4 3 3 1 1 3
## [2721] 3 1 4 1 1 4 3 3 1 4 4 4 4 3 1 1 4 1 1 1 1 3 4 4 3 2 3 3 3 2 3 2 3 3
## [2755] 2 4 3 4 4 3 3 3 3 1 3 4 3 3 3 3 3 4 4 4 4 2 1 4 3 1 1 1 1 4 4 4 4 3
## [2789] 2 4 2 2 4 4 4 4 3 3 3 4 3 4 3 4 4 2 4 3 4 4 3 1 3 1 1 1 3 4 1 4 3 3
## [2823] 3 4 4 4 4 4 4 1 1 1 1 1 1 3 1 1 4 1 1 1 1 1 1 1 1 1 1 4 4 1 1 4 1 4 1
## [2857] 1 4 4 4 4 4 4 3 3 4 3 3 1 1 1 1 1 4 1 1 1 1 1 2 2 3 3 2 3 3 2 4 1 3 3
## [2891] 3 3 4 3 2 2 2 2 1 3 3 3 1 1 1 4 1 2 1 1 3 1 1 1 1 1 3 3 3 3 3 1 1 3
## [2925] 4
##
## Within cluster sum of squares by cluster:
## [1] 3780.593 4358.496 5118.958 4385.936
##  (between_SS / total_SS =  45.1 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

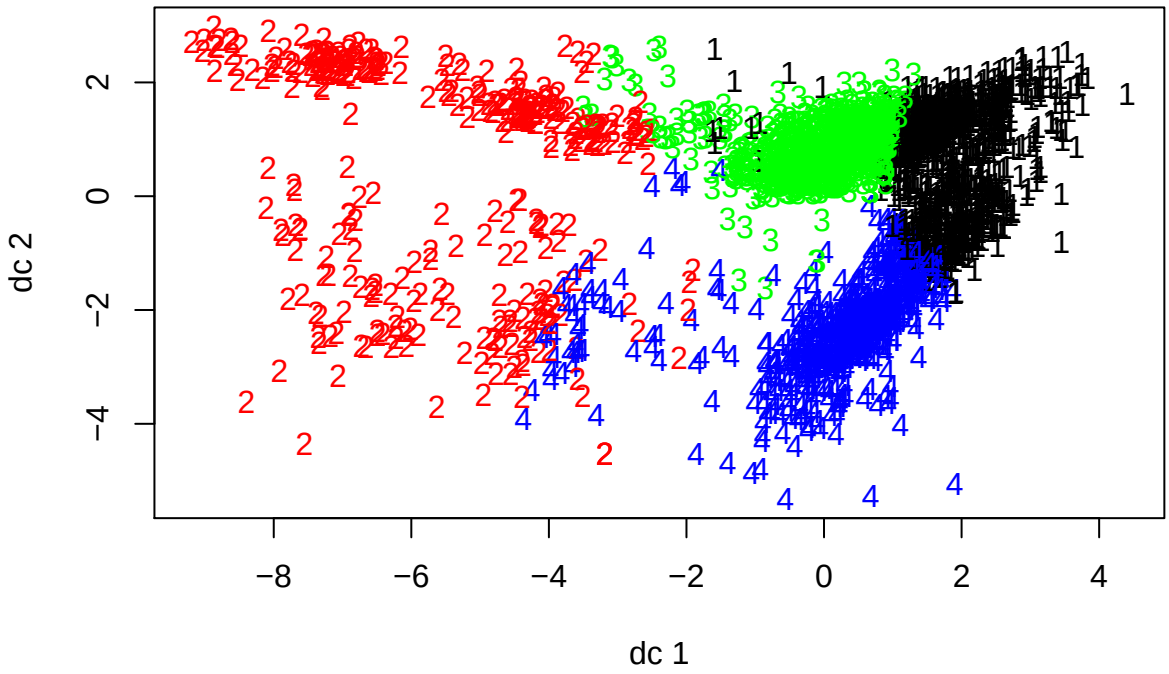
We group the data into four clusters, each designated by a circle. The clusters are distinct but they tend to intermingle with the other clusters, resembling a Venn diagram with overlapping areas. The leftmost cluster seems particularly dispersed. Reducing the clusters to three did not improve things much so we will stick with four PCs, despite the fact that the clustering seems to be falling apart.

```
clusplot(AmesPCAMeths, AmesCl$cluster, color = TRUE)
```



Finally, we group cluster members by color to get a better sense of how they disperse and how much they intermingle.

```
plotcluster(AmesPCAMeths, AmesCl$cluster)
```



The patterns appear pretty tight, except for the left-most cluster. A lot of variability (and so uncertainty) in the model resides there.