Predicting Sales Price of Houses

DSC 424 - Advanced Data Analysis

Dr. John McDonald

Presented By:

- Christian Craig
- Nina Eskandari
- Umair Chaanda
- Vineet Dcunha

Introduction

Housing prices are an important reflection of the economy, and housing price ranges are of great interest for both buyers and sellers. In this project, house prices will be predicted given explanatory variables that cover many aspects of residential houses.

As continuous house prices, they will be predicted with various analysis.

Exploratory Analysis

Dimensions & Detail

```
> nrow(housingTrain) # Report number of rows in dataset
[1] 1460
> ncol(housingTrain) # Report number of columns in dataset
[1] 81
```

Housing training set has 1460 rows and 81 columns.

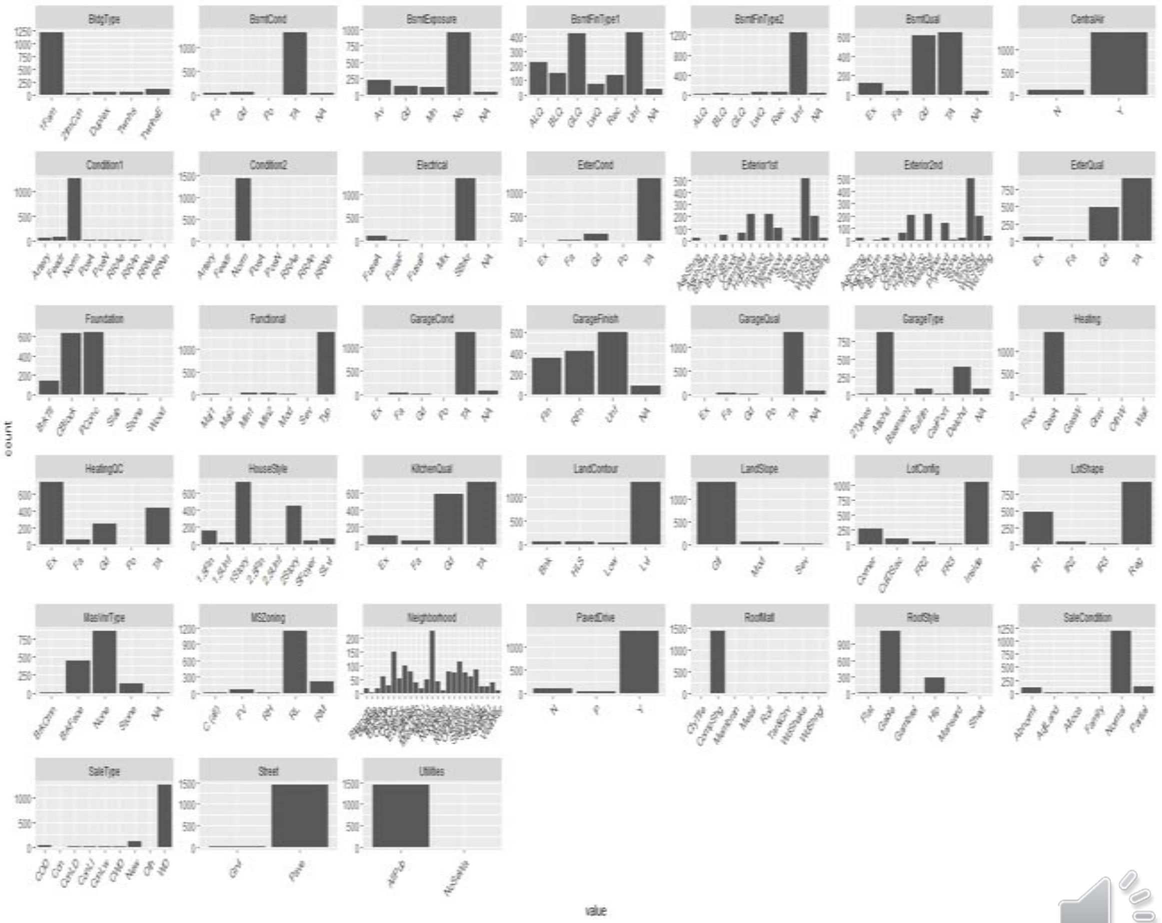
There are 38 numeric variables, and 42 categorical variables.

- NA's can be important factors
  - Ex: Pool v. No pool

```
> NA_Columns_Perc<-as.matrix(sort(colMeans(is.na(NA_Columns))))
> NA_Columns_Perc
                     [,1]
Electrical   0.0006849315
MasVnrType   0.0054794521
MasVnrArea   0.0054794521
BsmtQual     0.0253424658
BsmtCond     0.0253424658
BsmtFinType1 0.0253424658
BsmtExposure 0.0260273973
BsmtFinType2 0.0260273973
GarageType   0.0554794521
GarageYrBlt  0.0554794521
GarageFinish 0.0554794521
GarageQual   0.0554794521
GarageCond   0.0554794521
LotFrontage  0.1773972603
FireplaceQu  0.4726027397
Fence        0.8075342466
Alley        0.9376712329
MiscFeature  0.9630136986
PoolQC       0.9952054795
```

Numerical Variables

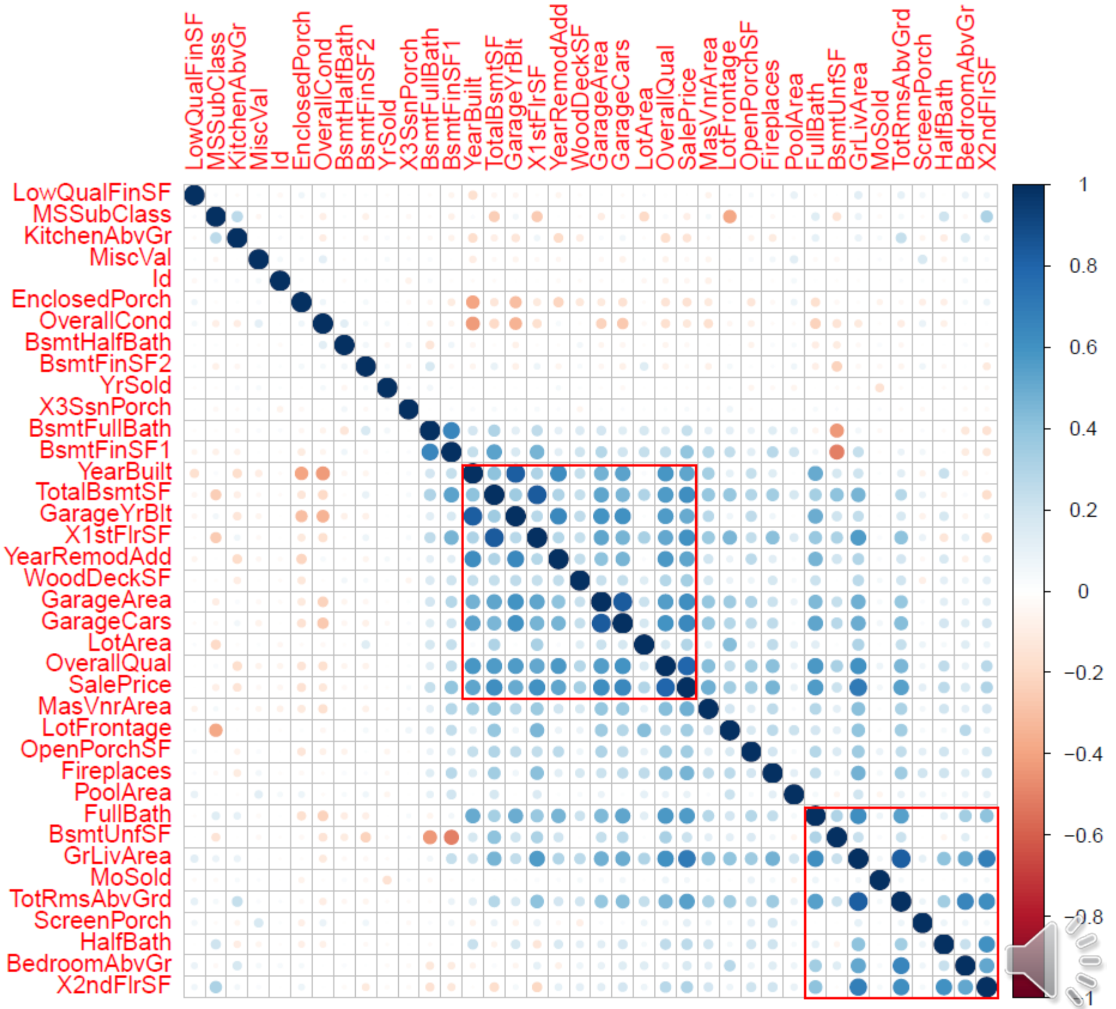
Categorical Variables



Correlation Matrix

A correlation matrix is a table showing correlation coefficients between sets of variables.

The following corrplot shows two main correlation factors between the variables.



Regularized regression

Regularized regression is the type of regression where the coefficient estimates are constrained to zero. The magnitude (size) of coefficients, as well as the magnitude of the error term, are penalized.

Complex models are discouraged, primarily to avoid overfitting.

Common types of regularized regression methods are Ridge regression, Lasso regression, and Elastic Net.

Regularized Regression

- **Parameter of Interest :** SalePrice

- **Feature Engineering:**
- Total number of bathrooms : FullBath + HalfBath*0.5 +BsmtFullBath + BsmtHalfBath*0.5
- Total square feet: GrLivArea+ TotalBsmtSF

- **Data Preparation for modeling:**
- Log transformation of response variable
- Removing outliers
- Label encoding

Changed some categorical variables which have numerical order to ordinal variables:
'Ex'=5,'Gd'=4,'TA'=3,'Fa'=2,'Po'=1,'None'=0

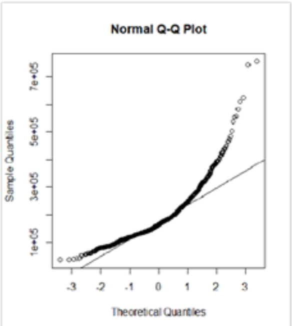
'BsmtQual', 'BsmtCond', 'GarageQual', 'GarageCond', 'ExterQual', 'ExterCond', 'HeatingQC', 'PoolQC', 'KitchenQual'

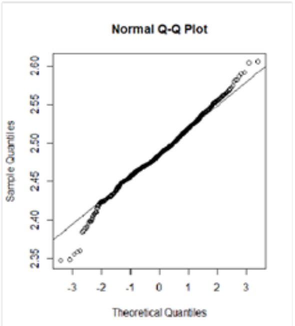
Sale price

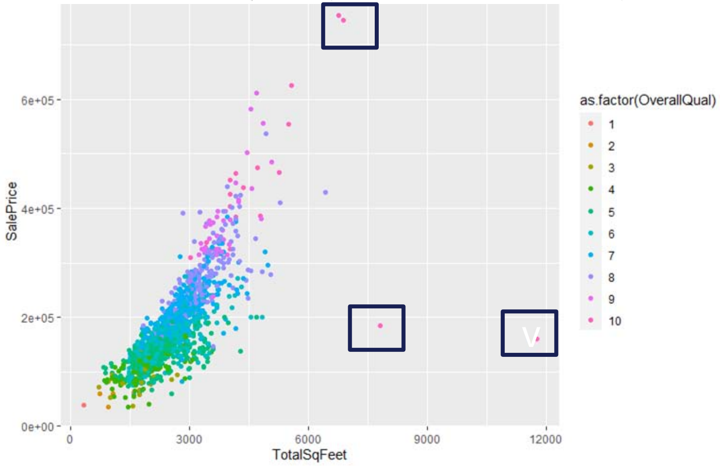
```
> skewness(housingtrain$SalePrice)
[1] 1.880941
> kurtosis(housingtrain$SalePrice)
[1] 9.509812
```

Log (Sale Price)

```
> skewness(housingtrain$SalePrice)
[1] -0.02035577
> kurtosis(housingtrain$SalePrice)
[1] 3.890477
```



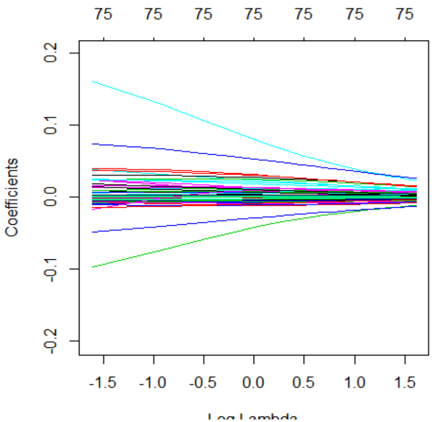


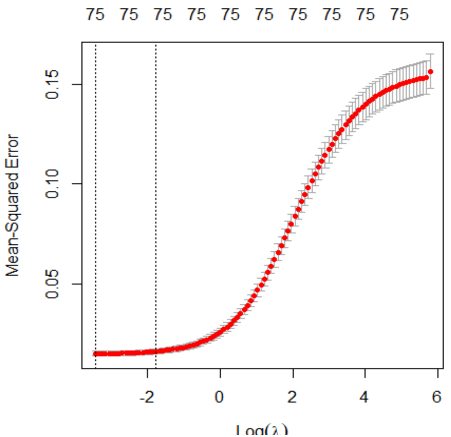


Regularized Regression

- **Ridge regression:**

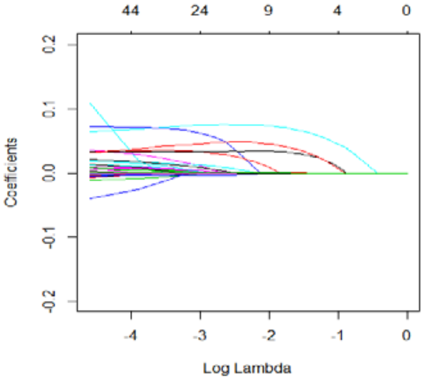
```
> RIDGEfit = cv.glmnet(xTrain,yTrain, alpha=0, nfolds=10)
> RIDGEfit$lambda.min
[1] 0.03239135
> RIDGEfit$lambda.1se
[1] 0.1728629

Call:  glmnet(x = xTrain, y = yTrain, alpha = 0, lambda = 0.1731081)

  Df  %Dev Lambda
1 75 0.906 0.1731

> rmseRidgetrain
[1] 0.1212295
```

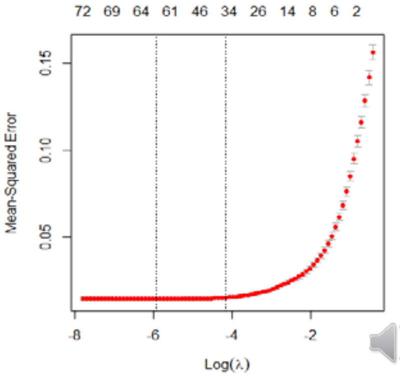




- **Elastic Net:**

```
> Elasticfit = cv.glmnet(xTrain, yTrain, alpha=0.5, nfolds=10)
> Elasticfit$lambda.min
[1] 0.00267683
> Elasticfit$lambda.1se
[1] 0.01567825

Call:  glmnet(x = xTrain, y = yTrain, alpha = 1, lambda = 0.01567825)

  Df   %Dev   Lambda
1 26 0.8937 0.01568

> rmseElastictrain
[1] 0.1203534
```





Lasso Regression

```
> LASSOfit = cv.glmnet(xTrain, yTrain, alpha=1, nfolds=10)
> LASSOfit$lambda.min
[1] 0.0003638599
> LASSOfit$lambda.1se
[1] 0.006508178
```

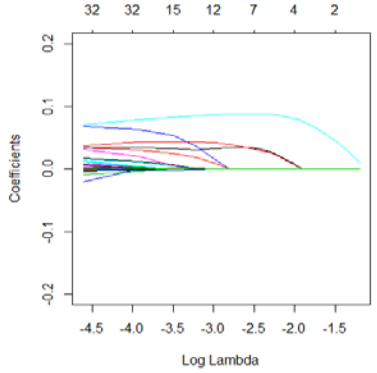
```
Call:  glmnet(x = xTrain, y = yTrain, alpha = 1, lambda = 0.006508178)

  Df   %Dev   Lambda
1 40 0.9093 0.006508
```

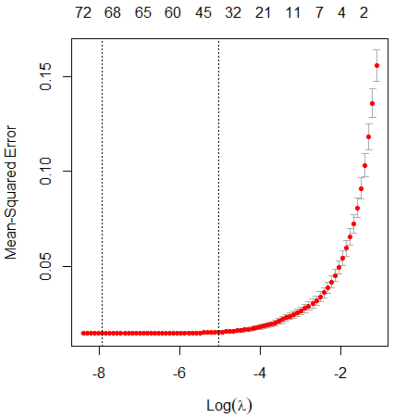
```
> rmseLASSOtrain
[1] 0.1191232
```

- **variable selection:**

Lasso regression selected 31 predictors out of the initial 72 variables using lambda.lse.

- **coefficients of variables:**





(Intercept)	8.1719097833
Street	0.0923383665
CentralAir	0.0696734348
OverallQual	0.0666697925
OverallCond	0.0357093970
GarageCars	0.0350524624
Fireplaces	0.0330508860
KitchenAbvGr	-0.0324500877
TotBathrooms	0.0311379158
SaleCondition	0.0207846754
KitchenQual	0.0174219549
PavedDrive	0.0169968409
Foundation	0.0136131080
Functional	0.0119864100
BldgType	-0.0107303256
BsmtQual	0.0096808355
MSZoning	-0.0065989332
HeatingQC	0.0061919783
ExterQual	0.0057845244
TotRmsAbvGrd	0.0053397712
Alley	0.0045977249
ExterCond	-0.0045443558
GarageCond	-0.0041228686
LotShape	-0.0038282446
GarageType	-0.0017781463
MasVnrType	0.0016805223
YearBuilt	0.0014170629
YrSold	-0.0011902375
FireplaceQu	-0.0010787650
BsmtExposure	-0.0010566657
YearRemodAdd	0.00075619[illegible]9

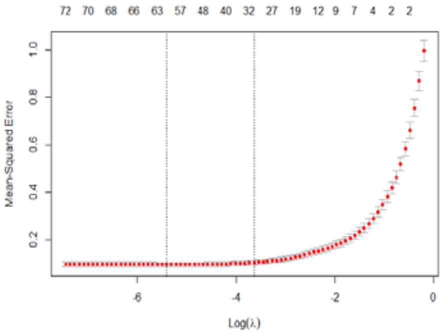
Summary Of Final Model

- **Final Model:** Lasso regression

- **Standardized data to get standardized Beta:**

```
xTrain <-scale(xTrain)
yTrain<- scale(yTrain)
```

```
LASSOfit$lambda.min
0.004471833
LASSOfit$lambda.1se
0.02619162
```

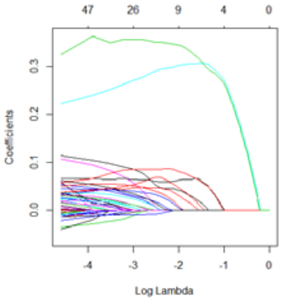


	Df	%Dev	Lambda
86	7	0.81740	0.15
87	9	0.82370	0.14
88	9	0.83040	0.13
89	10	0.83640	0.12
90	11	0.84220	0.11
91	12	0.84810	0.10
92	13	0.85420	0.09
93	15	0.86230	0.08
94	16	0.87070	0.07
95	19	0.87850	0.06
96	21	0.88620	0.05
97	26	0.89340	0.04
98	28	0.90040	0.03
99	33	0.90690	0.02
100	47	0.91360	0.01
101	75	0.91840	0.00

- **Increased lambda from lambda.1se to 0.1 :**
- Number of selected variables: 31 >> 12
- R square: 0.902 >> 0.848
- RMSE: 0.311 >> 0.389

- **Important variables:**
- Coefficient of variables included in the mode

Variable	standardized beta
TotalSqFeet	0.349245024
OverallQual	0.295903256
TotBathrooms	0.085753377
YearRemodAdd	0.064599293
GarageCars	0.058765700
YearBuilt	0.045731967
GarageArea	0.044595658
Fireplaces	0.030694746
CentralAir	0.023997537
BsmtFinSF1	0.007214968
GarageType	-0.004665385
SaleCondition	0.002253244



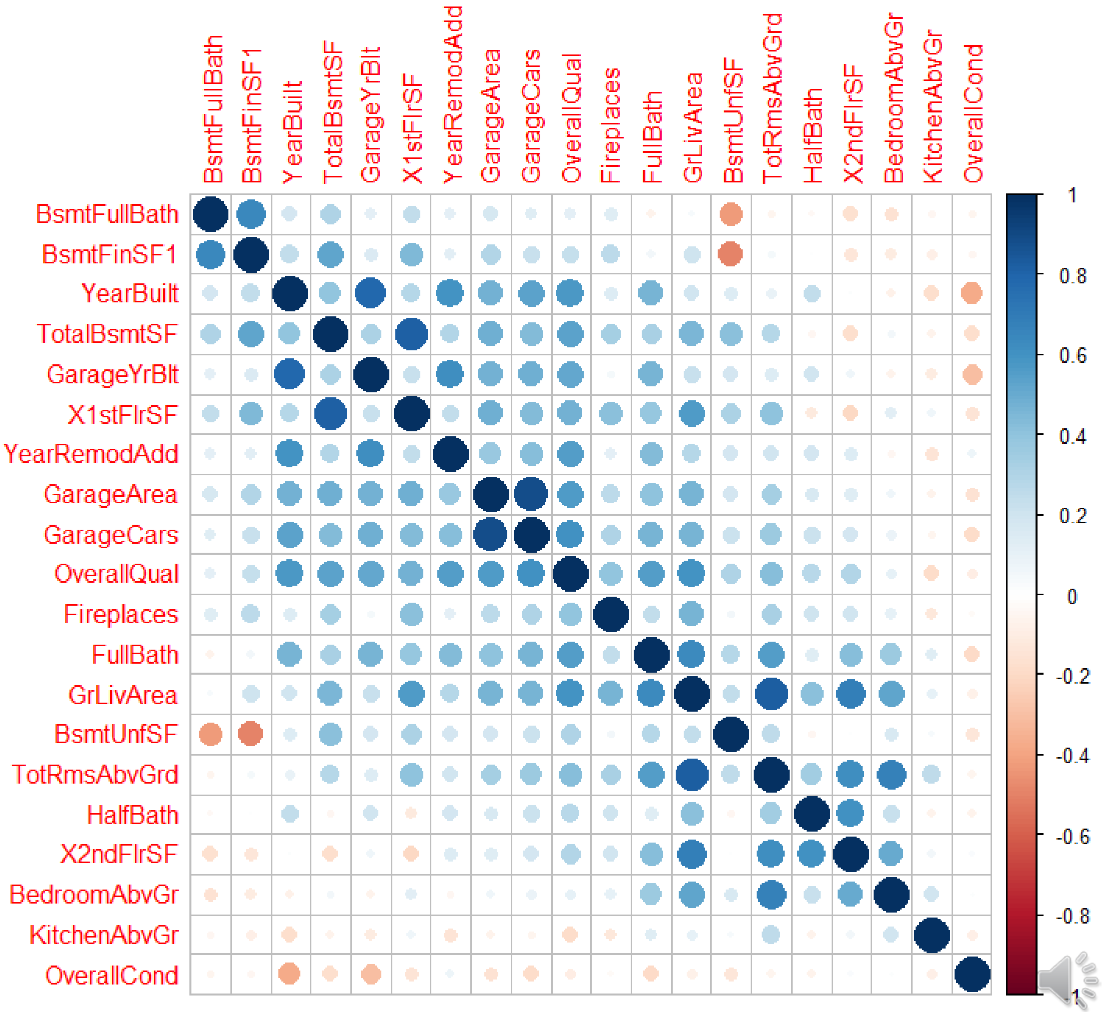
Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a statistical procedure that allows us to summarize the information contained in a large set by means of a smaller set of "summary indices" that can be more easily visualized and analyzed. It is a very common technique for "dimensionality reduction" and finding the "latent/hidden" factors from the data.

Testing Correlation Matrix

```
> # Compute the correlation matrix to see if there is significant
> # correlation to exploit
> corTests3 = cor(houseNum3)
>
> # Visualize correlation matrix
> cor.housing3 = cor(houseNum3, use="complete.obs")
> corrplot(cor.housing3, method="circle", order="AOE")
> houseCorrTest3 = corr.test(houseNum3, adjust="none")
> Mhouse3 = houseCorrTest3$p
> MTesthouse3 = ifelse(Mhouse3 < .01, T, F) # if Mhouse3 value <
> colSums(MTesthouse3) - 1
 OverallQual  OverallCond    YearBuilt YearRemodAdd   BsmtFinSF1
          19           12           18           18           15
BsmtFullBath     FullBath     HalfBath BedroomAbvGr KitchenAbvGr
          13           17           14           14           14

.01 then TRUE else FALSE

   BsmtUnfSF  TotalBsmtSF    X1stFlrSF    X2ndFlrSF    GrLivArea
          15           17           19           15           18
TotRmsAbvGrd   Fireplaces  GarageYrBlt   GarageCars   GarageArea
          16           16           17           18           17
```

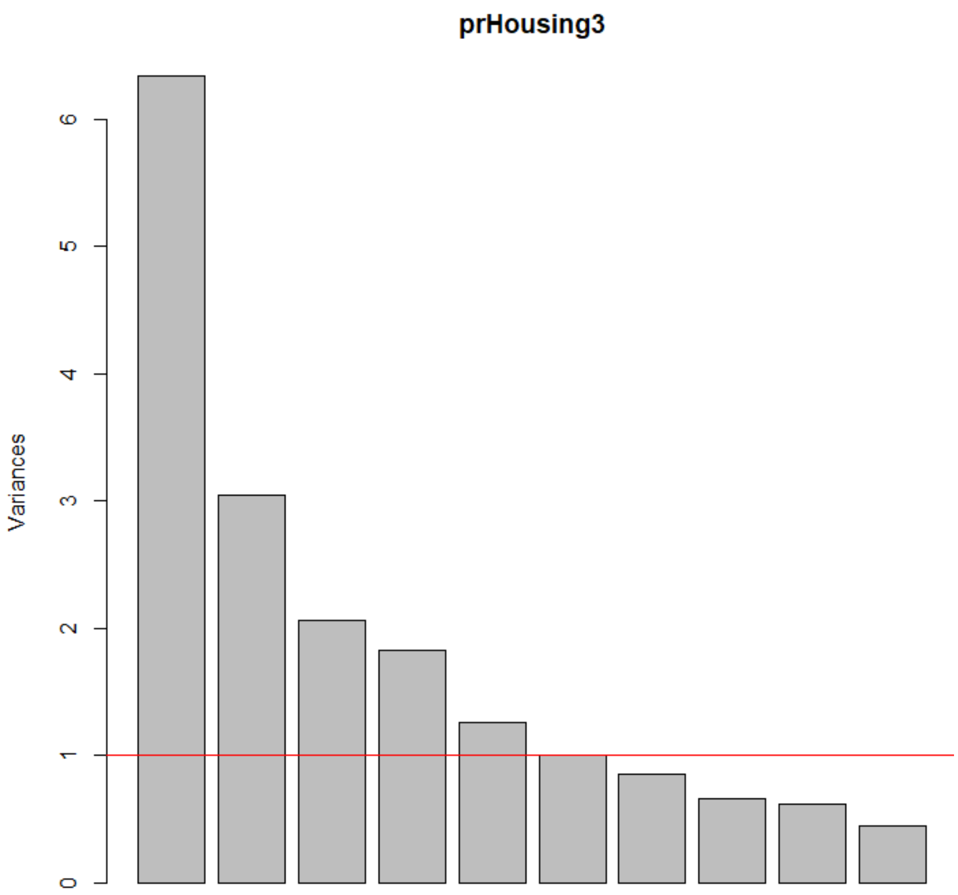
- None of the variables shows any problem.
- Most variables are positively correlated with each other.
- Seems some grouping among the variables.



"prcomp"

```
> #####################################################################
> # Compute "prcomp" with scaling/correlation matrix
> # and determine number of components
> #####################################################################
> prHousing3 = prcomp(houseNum3, scale=T)
> plot(prHousing3)           # The scree plot
> abline(1, 0, col="red")   # Put in a line at var=1
> summary(prHousing3)         # Get a summary including variances
Importance of components:
                         PC1    PC2    PC3     PC4     PC5     PC6     PC7
Standard deviation     2.518 1.7446 1.4351 1.35146 1.12379 0.99941 0.92469
Proportion of Variance 0.317 0.1522 0.1030 0.09132 0.06315 0.04994 0.04275
Cumulative Proportion  0.317 0.4691 0.5721 0.66344 0.72658 0.77652 0.81928
                          PC8     PC9    PC10   PC11    PC12    PC13    PC14
                       0.8111 0.78827 0.66865 0.6371 0.55536 0.52086 0.50884
                       0.0329 0.03107 0.02235 0.0203 0.01542 0.01356 0.01295
                       0.8522 0.88324 0.90560 0.9259 0.94131 0.95488 0.96782

                           PC15    PC16    PC17    PC18    PC19    PC20
                        0.46379 0.39146 0.36910 0.31021 0.19816 0.05863
                        0.01075 0.00766 0.00681 0.00481 0.00196 0.00017
                        0.97858 0.98624 0.99305 0.99786 0.99983 1.00000
```

- **Five** principal components required to explain more than **72%** of the variance for this data.
- At **PC5**, the Cumulative Proportion of variance is 0.72 (72%).
- The evening out pattern in the Scree Plot shows that after **five**, the components start containing unexplainable noise.
- **PC1** has the most Proportion of Variance.



Loadings from Principal

```
> ########################################################################
> # use "principal" to compute the Principal Component Analysis
> # with prHousing number of components, and with varimax factor rotation
> ########################################################################
> # prcomp is what we use to select our components
> # principal is used if we need to rotate the components
> principalHousing3 = principal(houseNum3, rotate="varimax", nfactors=5)
> # factors determined from prHousing scree-plot
> print(principalHousing3$loadings, cutoff=.5, sort=T)
```

Loadings from principal with rotate="varimax" and "nfactors=5" shows a very nice set of components with much better separations of variables.

- **RC1** is a mix of GARAGE + Age Of Property;
- **RC2** is mostly Above Ground + 2nd Floor;
- **RC4** is Basement + 1st Floor area;
- **RC3** is nothing but BASEMENT;
- **RC5** is a negative association between OverallCond and KitchenAbvGr;

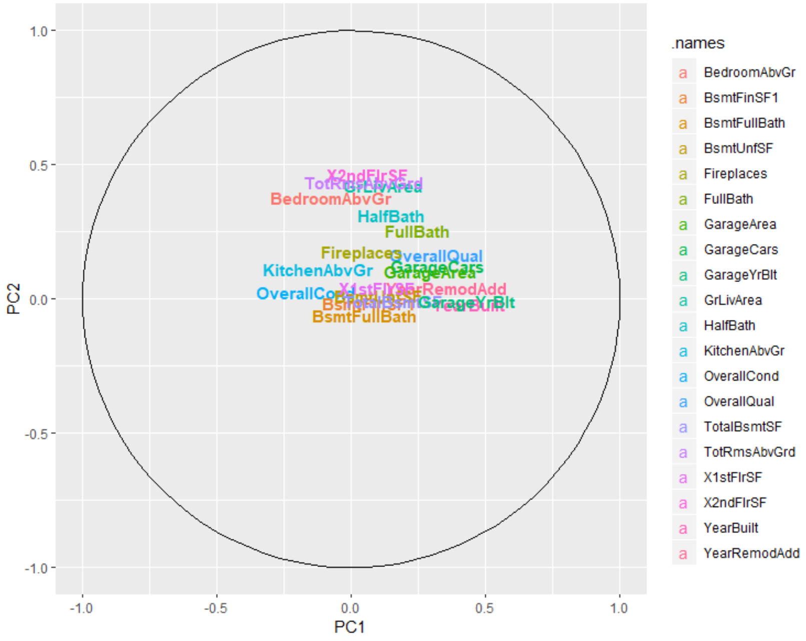
Loadings:

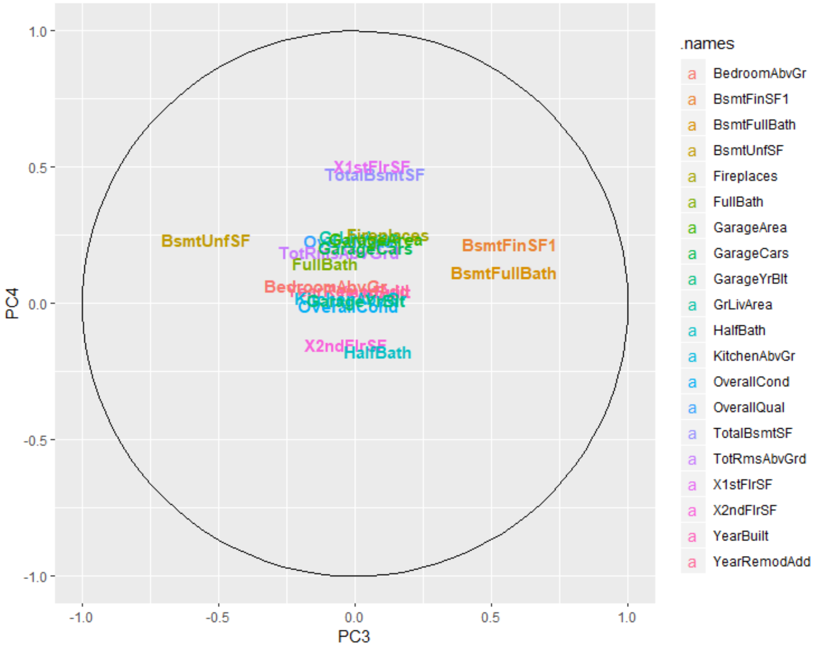
	RC1	RC2	RC4	RC3	RC5
OverallQual	0.655				
YearBuilt	0.893				
YearRemodAdd	0.725				
FullBath	0.506				
GarageYrBlt	0.879				
GarageCars	0.657				
GarageArea	0.604				
X2ndFlrSF		0.890			
GrLivArea		0.811			
HalfBath		0.600			
BedroomAbvGr		0.726			
TotRmsAbvGrd		0.834			
TotalBsmtSF			0.850		
X1stFlrSF			0.909		
BsmtFinSF1				0.834	
BsmtUnfSF				-0.801	
BsmtFullBath				0.804	
OverallCond					-0.638
KitchenAbvGr					0.701
Fireplaces					

	RC1	RC2	RC4	RC3	RC5
SS loadings	4.181	3.674	3.202	2.143	1.332
Proportion Var	0.209	0.184	0.160	0.107	0.067
Cumulative Var	0.209	0.393	0.553	0.660	0.727

PCA_Plot

```
> # produce a PCA_Plot_Psych plot of the contributions
> source("PCA_Plot.R")
> PCA_Plot_Psyc(principalHousing3)           # plot PC1 and PC2
> PCA_Plot_Psyc_Secondary(principalHousing3) # plot PC3 and PC4
```





Common Factor Analysis (CFA)

Common factor analysis extracts maximum common variance from all variables and puts them into a common score.

Factor Loadings

The five factors are contributing to around **63%** variance. The five factors can be named as following:

- **Factor1**: Total house Area
- **Factor2**: House Quality and Year Built
- **Factor3**: Non Living Area
- **Factor4**: Basement Area
- **Factor5**: Garage Area

There are absolutely no contribution from variables like OverallCond, FullBath, KitchenAbvGr, and Fireplaces.

Chi-square of ~0 so we reject the null hypothesis. This value is well below our α of 0.05, leading us to reject the null hypothesis that the model adequately fits the data.

```
> ##################################################################
> # And finally, COMMON FACTOR ANALYSIS and compare the two loadings
> ##################################################################
>
> factanalHousing = factanal(houseNum3, 5)
> print(factanalHousing$loadings, cutoff=.5, sort=T)

Loadings:
             Factor1 Factor2 Factor3 Factor4 Factor5
X2ndFlrSF     0.958
GrLivArea     0.845
HalfBath      0.570
BedroomAbvGr  0.560
TotRmsAbvGrd  0.734
OverallQual           0.546
YearBuilt             0.887
YearRemodAdd          0.645
GarageYrBlt           0.833
TotalBsmtSF                   0.756
X1stFlrSF                     0.964
BsmtFinSF1                            0.913
BsmtUnfSF                            -0.709
BsmtFullBath                          0.612
GarageCars                                    0.873
GarageArea                                    0.743
OverallCond
FullBath
KitchenAbvGr
Fireplaces

               Factor1 Factor2 Factor3 Factor4 Factor5
SS loadings      3.413   3.074   2.769   1.854   1.676
Proportion Var   0.171   0.154   0.138   0.093   0.084
Cumulative Var   0.171   0.324   0.463   0.556   0.639

Test of the hypothesis that 5 factors are sufficient.
The chi square statistic is 3741 on 100 degrees of freedom.
The p-value is 0
```

Correspondence Analysis

From implementing Correspondence Analysis, we can see which overall (house) condition corresponds most with each sale price class

CA: Sale Price Class and Overall Condition

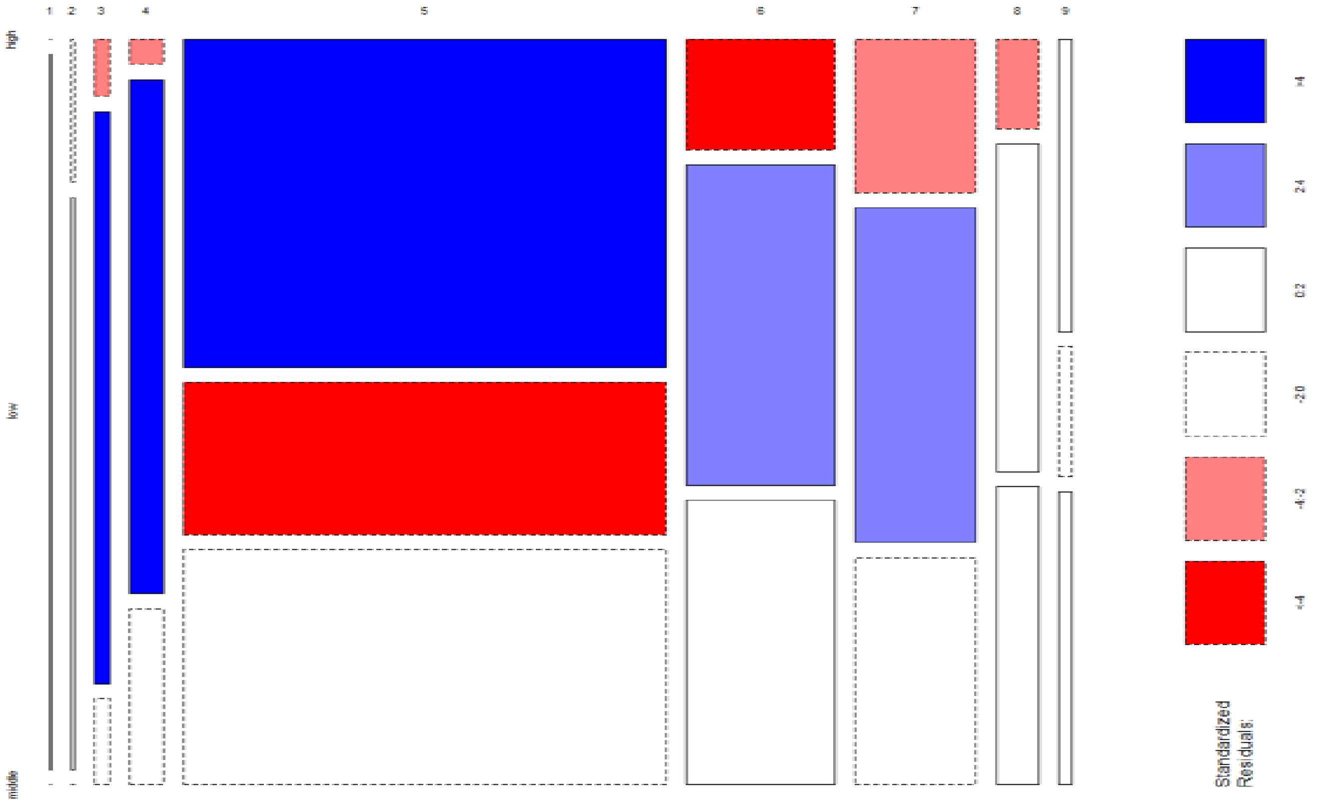
- Price Class broken into low, middle, and high
  - Near equal representation

```
# A tibble: 3 x 4
  PriceClassName count    low   high
  <fct>          <int>  <int>  <int>
1 low              487  34900 139600
2 middle           490 139900 190000
3 high             483 191000 755000
```

- Overall Condition
  - 10 Categories
    - 1 = Very Poor
    - 10 = Very Excellent
  - Can see if we can minimize this

```
  high low middle
1    0   1      0
2    1   4      0
3    2  20      3
4    2  41     14
5  377 175    269
6   39 113    100
7   44  96     65
8    9  33     30
9    9   4      9
```

CA: Sale Price Class and Overall Condition

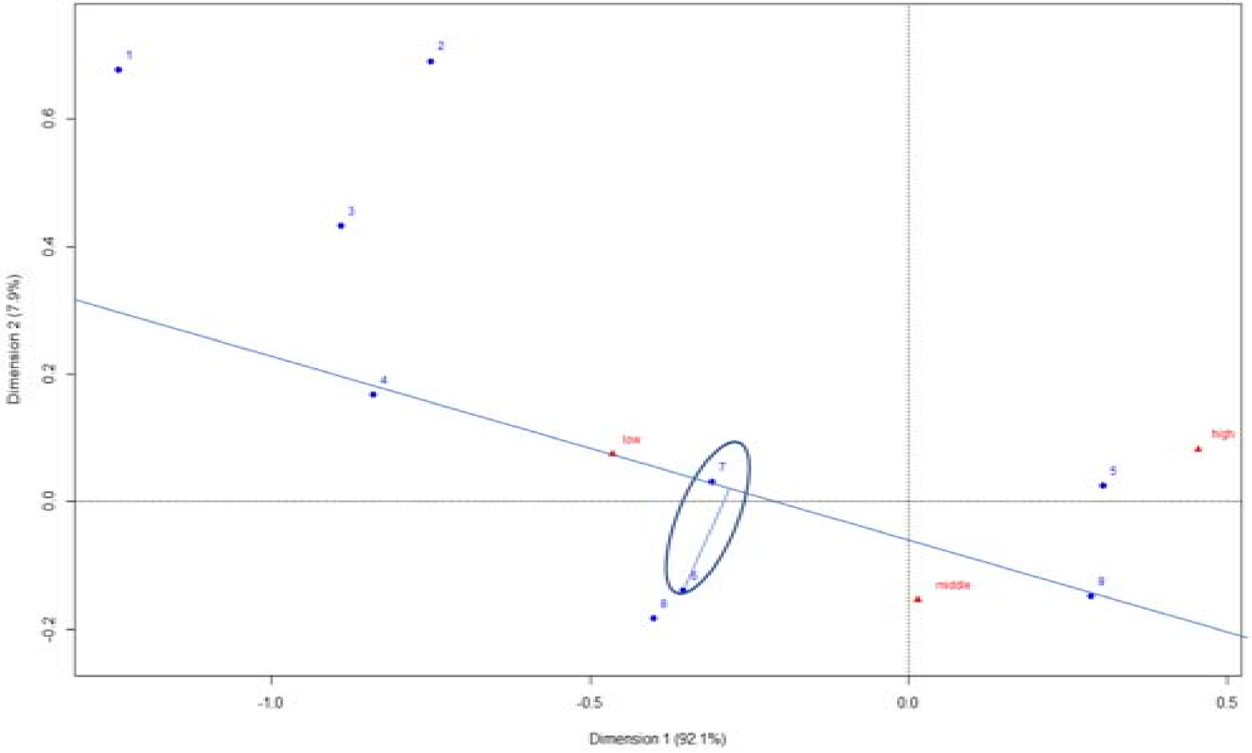


CA: Sale Price Class and Overall Condition

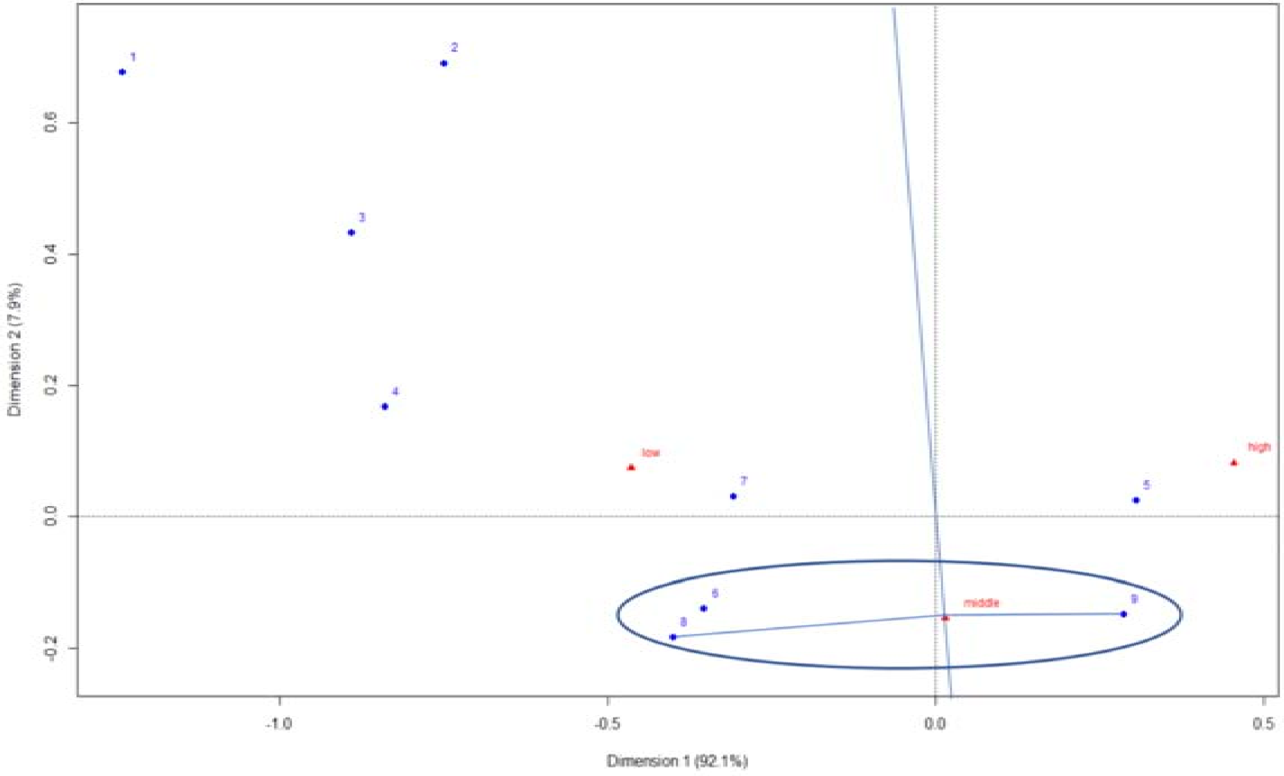
```
> summary(c3)

Principal inertias (eigenvalues):

 dim    value      %   cum%   scree plot
 1      0.140483  92.1  92.1  ***********************
 2      0.012064   7.9 100.0  **
        -------- -----
 Total: 0.152547 100.0
```

```
Rows:
    name   mass  qlt  inr    k=1 cor ctr    k=2 cor ctr
1 |    1 |    1 1000    9 | -1241 770   8 |  677 230  26 |
2 |    2 |    3 1000   23 |  -750 541  14 |  691 459 135 |
3 |    3 |   17 1000  110 |  -891 809  97 |  433 191 266 |
4 |    4 |   39 1000  188 |  -840 962 196 |  168  38  91 |
5 |    5 |  562 1000  345 |   305 993 372 |   25   7  30 |
6 |    6 |  173 1000  163 |  -354 866 154 | -139 134 278 |
7 |    7 |  140 1000   89 |  -309 990  95 |   31  10  11 |
8 |    8 |   49 1000   63 |  -401 828  56 | -183 172 136 |
9 |    9 |   15 1000   10 |   286 789   9 | -148 211  27 |

Columns:
    name   mass  qlt  inr    k=1 cor ctr    k=2 cor ctr
1 | high |  331 1000  462 |  454 969 486 |   82  31 183 |
2 |  low |  334 1000  485 | -465 975 513 |   74  25 153 |
3 | mddl |  336 1000   53 |   14   9   0 | -154 991 664 |
```

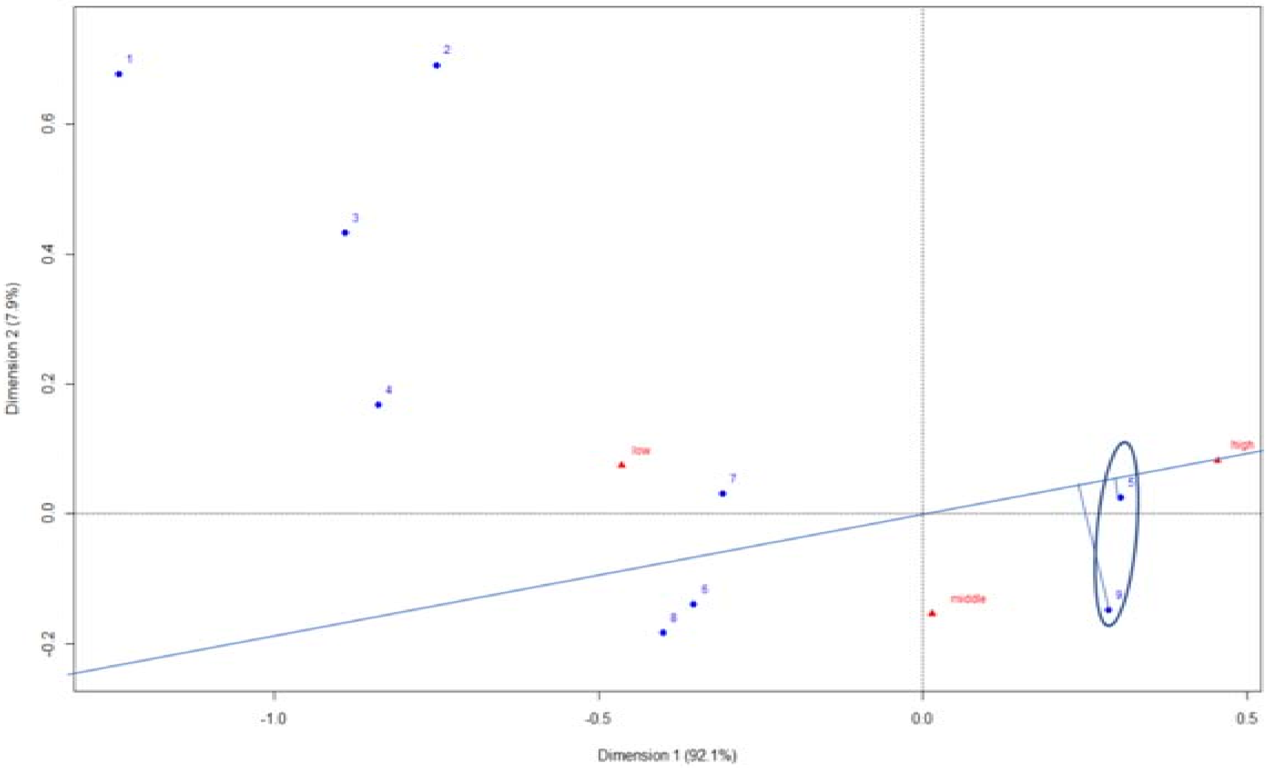
CA: Low Price Class



CA: Middle Price Class



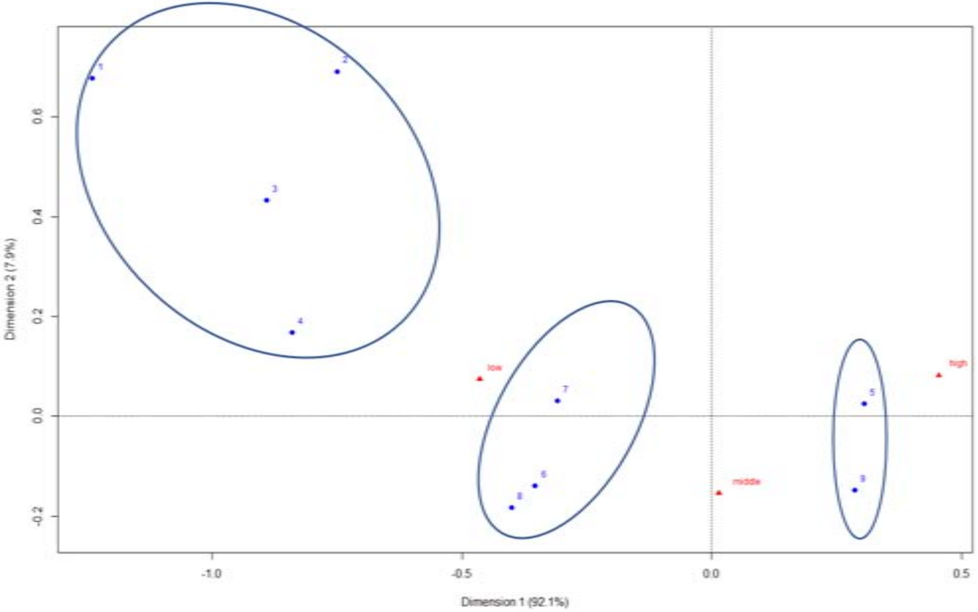
CA: High Price Class



CA: An Interesting Pattern

- Notice the groupings
  - [1,2,3,4]
  - [6,7,8]
  - [5,9]
- Begs the question "can we narrow down to 3 categories?"

- Need to look further into why 5 is grouped with 9
  - Class imbalance?




```
> table(housing_train$OverallCond)

  1   2   3   4   5   6   7   8   9 
  1   5  25  57 821 252 205  72  22 
```

Multiple Correspondence Analysis

The Multiple correspondence analysis (MCA) is an extension of the simple correspondence analysis for summarizing and visualizing a data table containing more than two categorical variables.

MCA - Sales Price

MCA for the following variables

- SalesClass
- YearRemodAddClass
- YearBuiltClass
- YrSold

Sales prices are high for the houses sold in the year 2007 and 2009. These houses are usually built and remodelled in Late 2000's. (Highlighted in Grey)

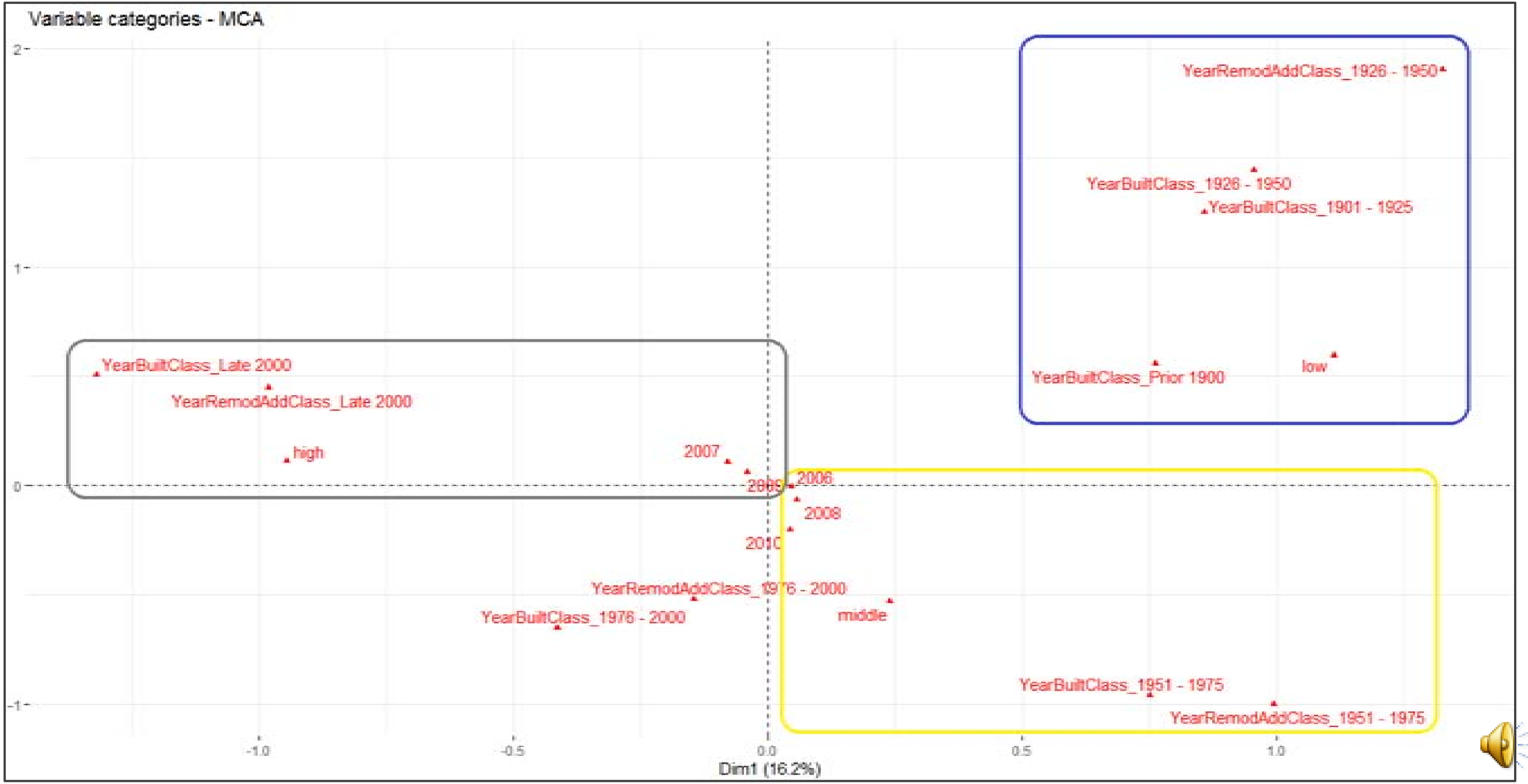
Houses sold in the year 2006, 2008 and 2010 have average sales price. The houses in this category were built and remodelled between the years 1951 - 1975(Highlighted in Yellow).

The year 2008 had a great economic recession. The slump in house prices can be attributed to this recession.

The houses built and remodelled prior to 1951 have low prices.

The houses built and remodelled between the year 1976 and 200 are plotted between high prices and average prices.

Plot - Sales Price



MCA - Sales Price (Extended)

MCA for the following variables

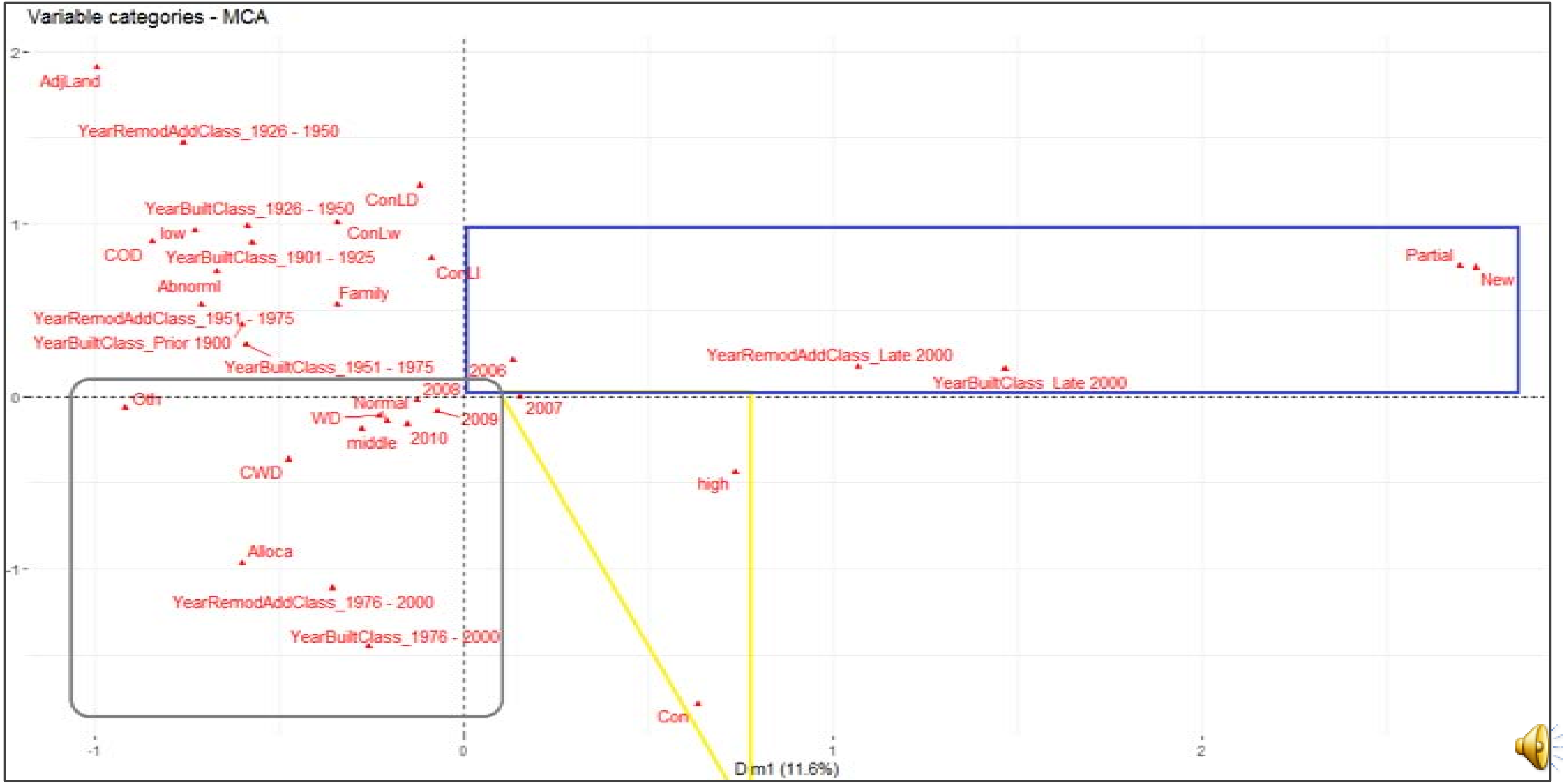
- SalesClass
- YearRemodAddClass
- YearBuiltClass
- SaleCondition
- SaleType
- YrSold

Sales prices are high for the houses sold in the year 2007 having the type of sale as Con 'Contract 15% Down payment regular terms'. (Highlighted in Yellow)

Houses sold in the year 2008, 2009, 2010 have average sales price. The type of sale is CWD 'Warranty Deed – Cash' and WD 'Warranty Deed – Conventional'. The houses with average Sales Price are usually remodelled between year 1976 – 2000 and newly built between year 1976 – 2000. The sale condition was Normal for this category. (Highlighted in Grey).

The houses built and remodelled before 1976 have low prices. The sale type of these houses was ConLw 'Contract Low Down and low interest', ConLI 'Contract Low Interest', ConLD 'Contract Low Down' and COD 'Court Officer Deed/Estate'. Most of the houses were sold between family members.

Plot - Sales Price (Extended)



Linear Discriminant Analysis & Multidimensional Scaling

LDA will be implemented to locate a new feature space in order to project the data in a format that maximizes separability between the classes

LDA will produce a confusion matrix which will be implemented for MDS - to see if we have similar roof style profiles

LDA: Parameter of Interest

- LDA Parameter of interest = Roofstyle
  - Adjusted to ordinal
    - Flat → 1
    - Gable → 2
    - Gambrel → 3
    - Hip → 4
    - Mansard → 5
    - Shed → 6

- Goal = Separate the groups as much as possible
  - Have the middle value withing groups cause separation so we can focus on each group

LDA: Roofstyle Output

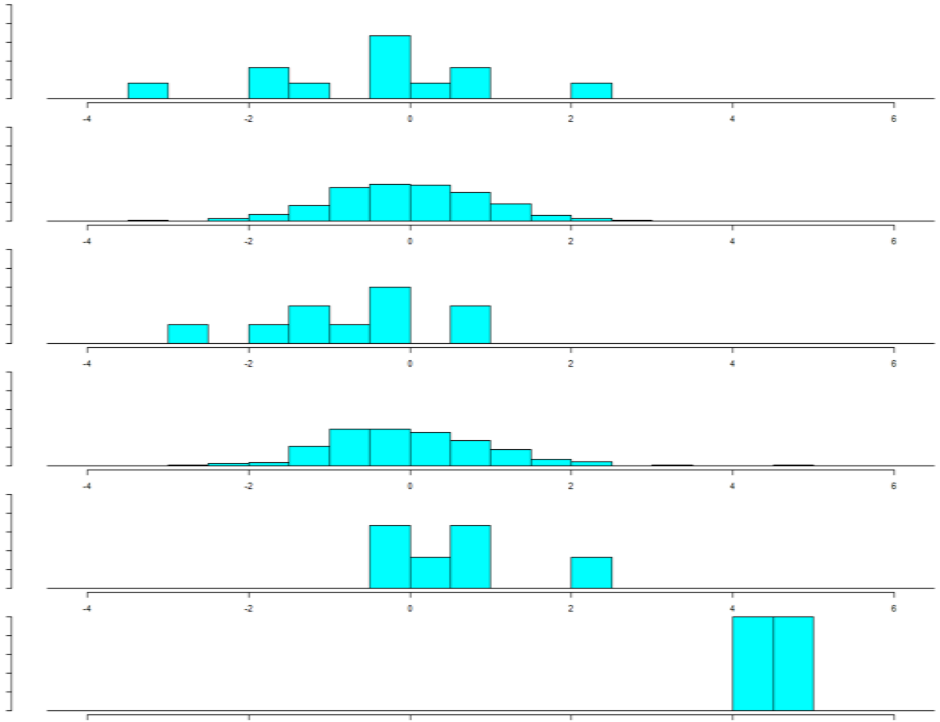
```
Coefficients of linear discriminants:
                       LD1           LD2           LD3           LD4           LD5
MSSubClass     3.053046e-03  1.668768e-03 -1.571022e-03 -1.655543e-03 -4.398299e-04
LotFrontage   -2.915105e-03  1.275523e-03 -1.584230e-03 -9.635418e-03 -9.702532e-03
LotArea        1.592742e-07 -2.800454e-05  1.356591e-05  8.563029e-06  3.676230e-06
OverallQual   -1.912185e-01  1.983061e-01  1.792856e-02 -1.536920e-01 -9.768921e-02
OverallCond    5.221281e-03 -1.391219e-01 -1.381019e-01  8.546249e-03  1.155574e-01
YearBuilt     -1.377996e-04 -1.501641e-02  7.879580e-03  8.073257e-04  3.436394e-02
YearRemodAdd   7.224063e-03 -4.173004e-04 -2.202367e-02 -1.534933e-02 -9.018419e-03
MasVnrArea    -2.467827e-03  1.831616e-03 -4.533538e-04 -1.193892e-03 -1.064628e-03
BsmtFinSF1     4.780128e-04  3.575601e-03 -1.224698e-03  8.733541e-04  1.037841e-03
BsmtUnfSF      1.034583e-03  3.160859e-03 -1.396777e-03  9.224614e-04  1.128152e-03
TotalBsmtSF   -3.964853e-04 -2.109536e-03  6.758098e-04 -3.139419e-04  7.161889e-04
X1stFlrSF     -3.870494e-03 -2.643083e-03 -2.604441e-03  1.245989e-03 -4.023123e-03
X2ndFlrSF     -1.794859e-03 -2.399462e-03 -3.221391e-03  7.484138e-04 -1.581301e-03
GrLivArea      2.544914e-03  4.536562e-04  2.851137e-03 -3.048879e-04  1.949531e-03
BsmtFullBath   1.904736e-01 -1.569381e-01  7.503917e-02  9.501496e-02  4.171679e-01
BsmtHalfBath   2.910284e-01 -5.600432e-04 -2.057832e-01  1.689020e-02 -3.750106e-01
FullBath       6.479728e-01  3.425695e-01  1.648439e-01 -5.899530e-01 -1.607102e-01
HalfBath       1.925144e-01  2.722065e-01 -2.915071e-01  1.433682e+00 -1.073133e+00
BedroomAbvGr   8.969979e-02  5.405714e-01  2.312889e-01 -6.518310e-01 -6.649403e-01
TotRmsAbvGrd  -1.545988e-01  1.780219e-01 -2.863631e-01  3.223963e-01  4.716609e-01
Fireplaces     7.135947e-02 -1.634243e-01  2.693130e-01 -2.563163e-01  7.582861e-02
GarageYrBlt    3.640107e-03  2.226029e-02  3.120707e-02  1.930813e-02 -2.802424e-02
GarageCars    -1.846648e-01 -3.747829e-01 -1.201215e-01 -2.049278e-02 -8.261702e-01
GarageArea     8.928013e-04 -3.376275e-04  7.281896e-05  7.546617e-04  1.056108e-03
WoodDeckSF    -3.778125e-04 -7.033140e-04  1.397417e-04  7.689830e-04 -1.307550e-03
OpenPorchSF   -2.166068e-04 -1.458660e-03  6.411148e-03 -4.673998e-03  1.090069e-03
X3SsnPorch    -1.646876e-03 -1.719891e-03  2.597904e-03 -1.242355e-03  6.584669e-04
MoSold        -4.525366e-03 -1.743000e-02 -5.391476e-02 -5.558448e-03  2.312056e-02
YrSold         4.897112e-03  8.611996e-03 -5.101012e-02  6.190944e-02  5.128039e-02
SalePrice     -3.852126e-06  7.698980e-07 -1.641925e-06 -3.106803e-06  2.736284e-06

Proportion of trace:
   LD1    LD2    LD3    LD4    LD5 
0.5096 0.1955 0.1441 0.0830 0.0678 
```

LDA: Roofstyle Scalings

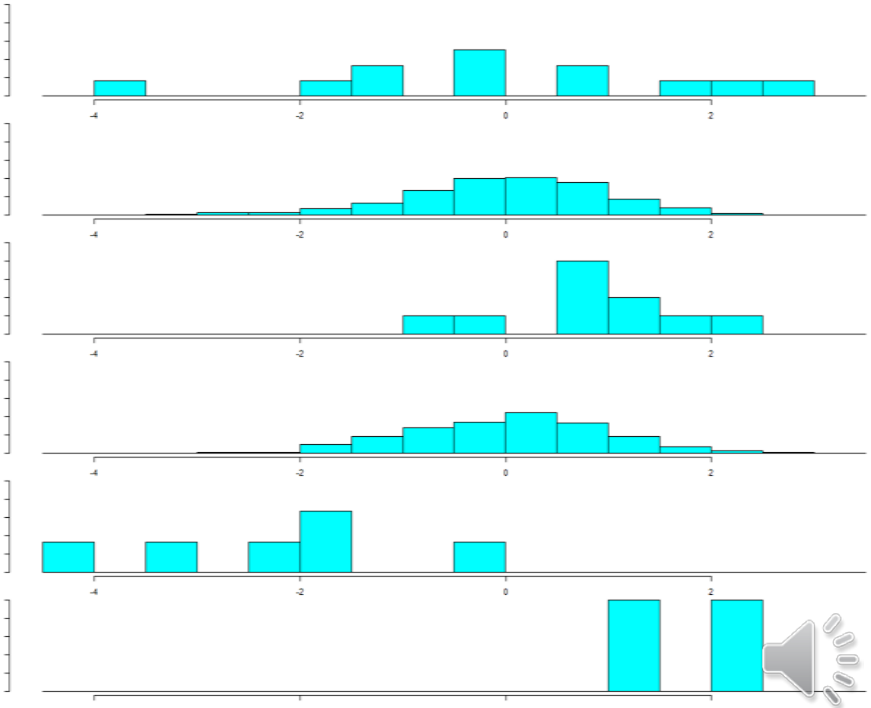
	LD1	LD2	LD3	LD4	LD5
GarageCars	-1.912185e-01	-3.747829e-01	-2.915071e-01	-6.518310e-01	-1.073133e+00
Fireplaces	-1.846648e-01	-1.634243e-01	-2.863631e-01	-5.899530e-01	-8.261702e-01
BsmtFullBath	-1.545988e-01	-1.569381e-01	-2.057832e-01	-2.563163e-01	-6.649403e-01
OverallCond	-4.525366e-03	-1.391219e-01	-1.381019e-01	-1.536920e-01	-3.750106e-01
MoSold	-3.870494e-03	-1.743000e-02	-1.201215e-01	-2.049278e-02	-1.607102e-01
YearBuilt	-2.915105e-03	-1.501641e-02	-5.391476e-02	-1.534933e-02	-9.768921e-02
X1stFlrSF	-2.467827e-03	-2.643083e-03	-5.101012e-02	-9.635418e-03	-2.802424e-02
X2ndFlrSF	-1.794859e-03	-2.399462e-03	-2.202367e-02	-5.558448e-03	-9.702532e-03
TotalBsmtSF	-1.646876e-03	-2.109536e-03	-3.221391e-03	-4.673998e-03	-9.018419e-03
X3SsnPorch	-3.964853e-04	-1.719891e-03	-2.604441e-03	-1.655543e-03	-4.023123e-03
OpenPorchSF	-3.778125e-04	-1.458660e-03	-1.584230e-03	-1.242355e-03	-1.581301e-03
WoodDeckSF	-2.166068e-04	-7.033140e-04	-1.571022e-03	-1.193892e-03	-1.307550e-03
BsmtHalfBath	-1.377996e-04	-5.600432e-04	-1.396777e-03	-3.139419e-04	-1.064628e-03
YearRemodAdd	-3.852126e-06	-4.173004e-04	-1.224698e-03	-3.048879e-04	-4.398299e-04
GarageArea	1.592742e-07	-3.376275e-04	-4.533538e-04	-3.106803e-06	2.736284e-06
LotArea	4.780128e-04	-2.800454e-05	-1.641925e-06	8.563029e-06	3.676230e-06
SalePrice	8.928013e-04	7.698980e-07	1.356591e-05	7.484138e-04	6.584669e-04
GrLivArea	1.034583e-03	4.536562e-04	7.281896e-05	7.546617e-04	7.161889e-04
LotFrontage	2.544914e-03	1.275523e-03	1.397417e-04	7.689830e-04	1.037841e-03
MSSubClass	3.053046e-03	1.668768e-03	6.758098e-04	8.073257e-04	1.056108e-03
MasVnrArea	3.640107e-03	1.831616e-03	2.597904e-03	8.733541e-04	1.090069e-03
BsmtUnfSF	4.897112e-03	3.160859e-03	2.851137e-03	9.224614e-04	1.128152e-03
BsmtFinSF1	5.221281e-03	3.575601e-03	6.411148e-03	1.245989e-03	1.949531e-03
YrSold	7.224063e-03	8.611996e-03	7.879580e-03	8.546249e-03	2.312056e-02
GarageYrBlt	7.135947e-02	2.226029e-02	1.792856e-02	1.689020e-02	3.436394e-02
TotRmsAbvGrd	8.969979e-02	1.780219e-01	3.120707e-02	1.930813e-02	5.128039e-02
OverallQual	1.904736e-01	1.983061e-01	7.503917e-02	6.190944e-02	7.582861e-02
HalfBath	1.925144e-01	2.722065e-01	1.648439e-01	9.501496e-02	1.155574e-01
FullBath	2.910284e-01	3.425695e-01	2.312889e-01	3.223963e-01	4.171679e-01
BedroomAbvGr	6.479728e-01	5.405714e-01	2.693130e-01	1.433682e+00	4.716609e-01

LDA: Separation Visualization

```
> ldahist(data = housingtrain.lda.values$x[,4],g=housing_train.edit$RoofStlyeNum)
```



```
> ldahist(data = housingtrain.lda.values$x[,5],g=housing_train.edit$RoofStlyeNum)
```



LDA: Confusion Matrix

- Some misclassification in row 4
  - Misclassified as group 2

- Can run MDS by Roofstyle to look further into this!

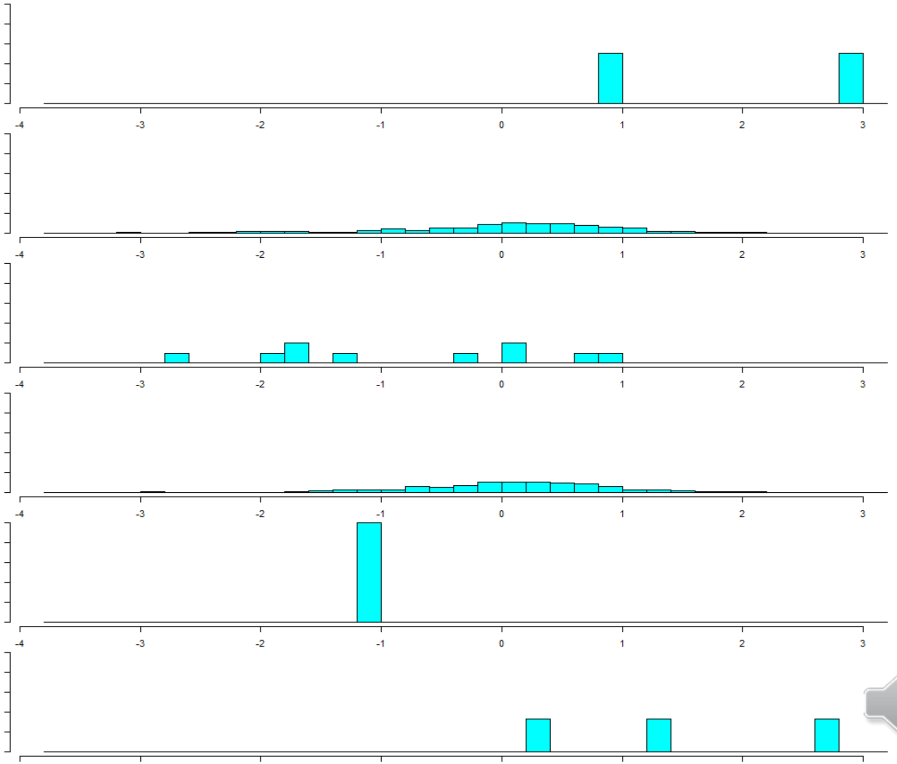
	1	2	3	4	5	6
1	6	5	0	0	1	0
2	14	990	14	35	7	3
3	0	5	5	0	0	0
4	9	192	5	71	0	1
5	0	3	0	0	3	0
6	0	0	0	0	0	2

LDA: Performed on The Test Set

- Notice Separation still with 5 &6
- Still Some misclassifications:
  - Group 4 Misclassified as group 2
  - Groups 5 &6

	1	2	3	4	5	6
1	1	1	0	0	0	0
2	19	859	17	22	3	0
3	0	9	1	0	0	0
4	7	130	2	69	2	0
5	0	1	0	0	0	0
6	1	2	0	0	0	0

```
> ldahist(data = housingtrain.lda.values$x[,3],g=housing_test.edit$RoofStlyeNum)
```



MDS: Roofstyle

- Goodness of fit
  - May need to look further into

- Stress = .00858
  - Very good

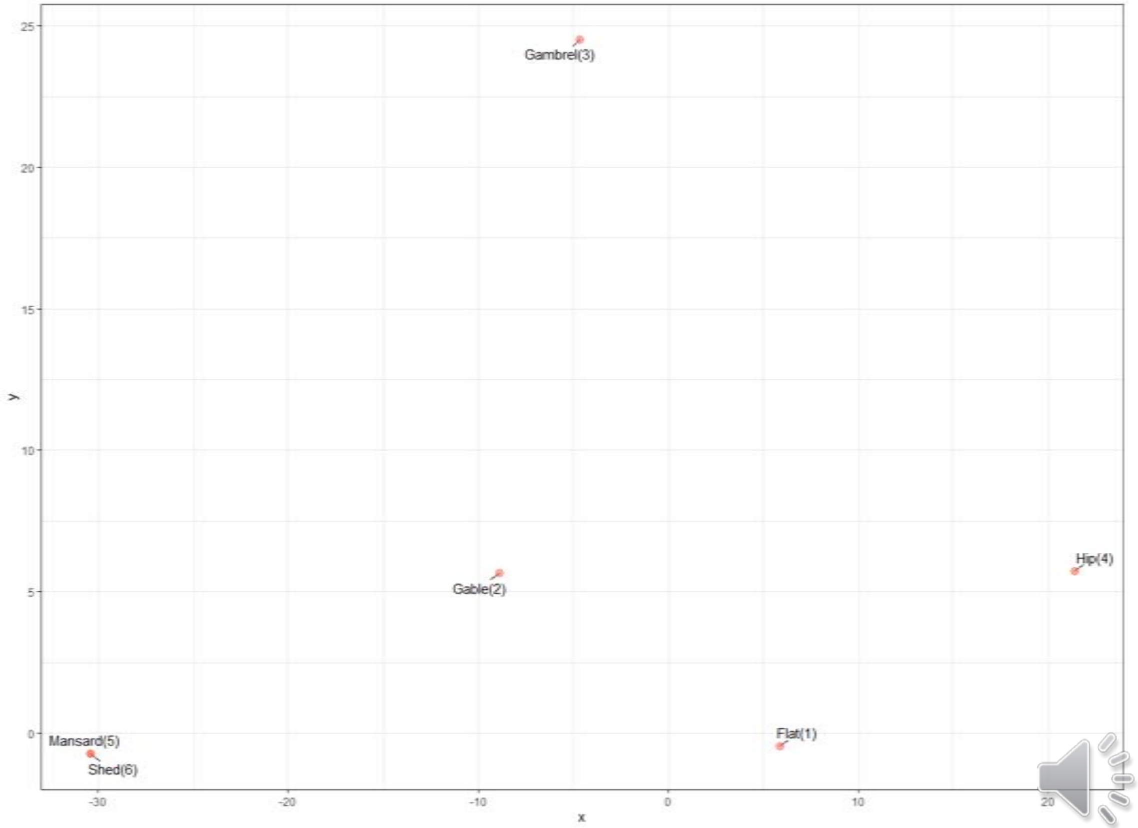
```
> fit <- cmdscale(Rooftbl,eig=TRUE, k=2) # k is the number of dim
> fit
$points
          [,1]       [,2]
1   5.891961 -0.4691393
2  -8.901380  5.6531402
3  -4.639301 24.5064059
4  21.393551  5.7322255
5 -30.401516 -0.7155886
6 -30.394414 -0.7366442

$eig
[1]  2.441229e+03  6.666551e+02 -4.541624e+00 -6.782711e+02 -5.036258e+03 -4.046696e+05

$x
NULL

$ac
[1] 0

$GOF
[1] 0.007516108 1.000000000
```

```
> roof.mds<-isoMDS(d)
initial  value 0.157686
iter   5 value 0.138204
iter  10 value 0.034102
iter  15 value 0.012837
iter  15 value 0.008584
final  value 0.008584
converged
> roof.mds$stress #very good
[1] 0.008583677
```

MDS: RoofStyle

- Group RoofStyle 5 & 6 together

- Notice RoofStyle 2 & 4 at the same point in y-axis



Thank you!