DSC 424 Advanced Data Analysis
Autumn 2019

Final Project

Multivariate Analyses
on Melbourne Housing Data

Ying Kam Chiu

A. Introduction

Home prices in Australia have lately become significantly overpriced and some claims are due for a significant downturn or collapse. Some have characterized this as the "Australian housing bubble," with Melbourne at its center. This significant upturn in price and increased interest in housing prices bears analysis. Particularly, it is worth examining what contributes to particular housing prices and how one can predict which houses appear in which neighborhoods. Here, I studied a Melbourne housing dataset, which contains 21 variables and 34,857 rows. During data preprocessing, some of the variables which could not be used directly were changed into usable forms. For instance, suburb names were transformed into four different suburb numbers, which were labeled as sublevel. To perform this transformation, suburbs were divided into four levels according to how they ranked in relation to Melbourne's most livable suburbs rankings (1 = highly ranked suburbs, 2 = upper-middle-ranked suburbs, 3 = lower-middle-ranked suburbs, and 4 = low-ranked suburbs). After data preprocessing process, 1109 data points were randomly selected for analysis.

In my analyses, I applied the following techniques: Correspondence Analysis and Linear Discriminant Analysis. I discovered four major factors for understanding house pricing and obtained four models for predicting house price or classifying property type.

B. Exploratory Analysis

1. Correlations and multicollinearity are discovered

During the exploratory analysis, I noted weak correlations between Price and multiple other variables, such as Rooms, Bedrooms, Bathroom, BuildingArea, and Sublevel, among other variables. Also, many variables except Price are correlated, such as Distance and sublevel, Rooms and THouse, BuildingArea and Rooms, Tunit and rooms and so on. Some extreme correlated pairs are Bedroom2 and Bathroom, suburb1 and sublevel, Tunit and Thouse.

2. Correlation Matrix for observing class separation

A correlation matrix was created to observe the classifications of the original dataset and explore the relationship of property type against all other independent variables. From the matrix, there is no clear separation of groups (figure 2.5).

C. Correspondence Analysis and Linear Discriminant Analysis

1. Application of Correspondence Analysis

Correspondence Analysis was used to explore the relationship between two categorical variables: property type and sublevel, in which we divided all suburbs into 4 levels, where and sublevel 4 contains the best suburbs while sublevel 1 is the least. To visualize the two variables, a mosaic plot was created by dividing first into horizontal bars whose widths are proportional to the probabilities associated with property types (figure 2.1). Then each bar was split vertically into bars that were proportional to the conditional probabilities of sublevel (suburb levels). From the interpretations of the mosaic plot and scale table (figure 2.2), we see that the most frequent property type is "h" in sublevel 1 while the least frequent is property type "t" in sublevel 4. Red tiles indicate significant negative residuals and blue tiles indicate significant positive residuals. The intensity of the color represents the magnitude of the residual. Property type "u" in sublevel 1 is represented by a dark blue tile, indicating very strong positive residuals. Property type "h" in sublevel 4 is represented by a light blue tile, indicating fairly strong positive residuals. Property type "h" in sublevel 1, and "u" in both sublevels 3 and 4 show on a light red tile, indicating fairly strong negative residuals. The rest are represented by white tiles, indicating weak association.

2. Residual analysis/model diagnostics

Chi-squared goodness of fit test was performed on CA based on the 2 hypotheses: null hypothesis: two variables are independent and alternative hypotheses: two variables are not

independent. The test result shows that the Chi-squared is 60.64 and the p-value of Chi-squared (3.33e-11) is much less than 0 (figure 2.3). At a confidence level of 95%, the p-value is smaller than the threshold and therefore we can reject the null hypothesis and accept the alternative hypothesis that two variables are associated in some ways. As a result, Chi-squared test has echoed with mosaic plot that house type and sublevel are associated.

3. Analysis of the results and discussion

From CA plots (figure 2.4), the levels of the property type and sublevel are ranked along the two dimensions and plotted in a projection graph of CA. The origin is the "average" mark. Sublevels 2, 3 and 4, to the right of origin, indicate a frequency more than the overall expected average frequency. On the other hand, sublevel 1 indicates a frequency less than the overall expected average frequency.

To create a property type "h" scale, we draw a line through the origin and "h". For each sublevel, we draw a perpendicular line to the scale. The closer the perpendicular line strikes the line that passes through the origin, the more this sublevel corresponds to property type "h". From the plot (figure 2.4), we can see that sublevels 1 and 4 correspond less to "h", while sublevel 2 corresponds more. In other words, there are probably more "h" in sublevel 2 and fewer in sublevels 1 and 4. Likewise, there are more "t" in sublevel 2 and less in sublevel 4; more "u" in sublevel 1 and less sublevel 3.

4. Application of Linear Discriminant Analysis (LDA)

In addition to CA, LDA is also performed with property type as a parameter of interest to find the best linear discriminant to separate the property types. The full model was then taken to run an initial LDA (figure 2.6). The prior probabilities of groups suggest that class "h" of 72.19%, "t", 8.03% and "u", 19.77%. LDA suggests two linear discriminant components:

*Full model LD1 = -.97*Price – .40*Rooms – .02*Distance – .15*Bedroom2 – .10* Bathroom – .08*Car – .25*Landsize + .01*BuildingArea + .10*Propertycount – .02*Years – .43*sublevel – .14*VendorBid + 1.18*Auction + .09*MethodS – 1.88*MethodSA + .26* MethodSP + .37*MethodVB*

*Full model LD2 = -.52*Price – .41*Rooms + .02*Distance + .54*Bedroom2 – .13* Bathroom – .26*Car – .26*Landsize + .00*BuildingArea – .03*Propertycount + .03*Years + .14*sublevel + .75*VendorBid – 30*Auction – .07*MethodS – .81*MethodSA – .18* MethodSP – .90*MethodVB*

In full model LD1, MethodSA weighs the highest, meaning it has the largest impact on the model. Bathroom, buildingArea, propertyCount, auction, methodS, methodSP and methodVB have positive impacts on the model, while the rest have negative impacts. In full model LD2, methodSA also weighs the highest, meaning it has the biggest impact on the model. Distance, bedroom2, car, buildingArea, Years, sublevel and vendorBid have positive impacts on the model, while the rest have negative impacts. LD1 explains 93.42% pf the variation of the data while LD2 only explains 6.58%.

5. Residual analysis/model diagnostics

The discriminant histogram (figure 2.7) of LD1 shows that "h" segregates itself quite well from "u", but there are some confusions in the middle. The discriminant histogram of LD2 does not segregate well with many confusions. The scatter plot also shows the classifications are confused (figure 2.8). The accuracy of the full model is 83.73% (figure 2.9). The misclassification rate is 16.27%: 50 "h" were misclassified as "t"; 18 "u" were misclassified as "h" etc. After running the initial LDA, we split the dataset into training (80%) and testing (20%) sets and tried to build a final predictive model manually by running LDA on the independent variable individually and ranking their accuracy rates

from the results (figure 2.10). The results show that Landsize, Rooms, Price, Distance and Bathroom are the most important variables to the model. The result of LDA on training set shows that the prior probabilities of groups indicates class "h" of 71.52%, "t", .08% and "u", 20.10% (figure 2.11).

LDA suggests two linear discriminant components:

*Final model LD1 = -.32*Landsize – .67*Rooms – .1.15*Price – .04*Distance + .44* Bathroom*
*Final model LD2 = -.32*Landsize + .88*Rooms + .90*Price + .03*Distance – .1.71* Bathroom*

6. Analysis of the results and discussion

In final model using train set, price weighs the highest in both LD1 and LD2, meaning it has the biggest impact on the model. In LD1, number of bathrooms has a positive impact on the model, while the rest have negative impact. In LD2, number of bathrooms and landsize have positive impact on the model, while the rest have negative impact. LD1 explains 96.91% of the variation in the data while LD2 only explains 3.09%. The discriminant histograms are similar to that of the initial LDA, LD1 shows "h" segregates itself quite well from "u", but there are some confusions in the middle (figure 2.13). The discriminant histogram of LD2 does not segregate well with many confusions. The scatter plot also shows the classifications are confused (figure 2.14). The accuracy of the model is 82.6%. The misclassification rate is 17.4%: 54 "h" misclassified as "t", 12 "misclassified as "h" etc (figure 2.12).

Next, we tested the performance of the model using the test set. The discriminant histograms LDA using test set are consistent with that of initial LDA and LDA using train set (figure 2.15). LD1 shows "h" segregates itself quite well from "u", but there are some confusions in the middle. The discriminant histogram of LD2 does not segregate well with many confusions. The scatter plot also shows the classifications are confused (figure 2.17). The accuracy of the final model is 83.59% and therefore the model is validated because the performance using test set is better than that of the training set (figure 2.15). The misclassification rate is 16.41%: 9 "h" misclassified as "t", 6 "u" misclassified as "h" and all "t" was misclassified.

To predict with five independent variables: landsize, rooms, price, distance and bathroom, properties, which were suggested by the manual method selection, the optimal predicted classifications will be 87.18% as "h", 12.54% as "t" and 0.28% as "u" (figure 2.18).

D. Conclusion

In our analyses on the housing data of the Greater Melbourne area, we found that type, which mainly involves property type and land size, most heavily influences price. Specifically, single home as house type, usually with larger land sizes, increase a home's price, whereas residence types apartment or condo unit adversely influences house price. A second important aspect is location, involving a residence's distance to downtown and the ranking of the suburb in which the property is located. Suburbs with lower suburb number (i.e. suburb 1 vs. suburb 4) corresponding with "better" places to live and vice versa. Living space and facilities also affect house prices positively: as living spaces become larger, price rises, and as the number of facilities increases, so do prices. The former includes building area and the number of rooms, and the latter refers to the number of bathrooms and parking space.

Additionally, we found that these predictors' influence on price differs depending on which quantile at which they are being regressed. For these data specifically, regressing by quantile appears to most accurately predict an outcome variable than ordinary least squares regression or regressing factor analysis scores by quantile. Notably, the implications for buyers with different budgets varies.

Looking at the data above, we see that bathrooms increases home price for bedrooms, especially for the .75 quantile. Therefore, if buyers want a large home and have many financial resources, this model may suggest that they look for a home with more bedrooms but not necessarily as many bathrooms. Sellers might also be advised to be aware that having extra bathrooms increases home value above and beyond bedrooms, especially if they have a home at the .75 quantile. Furthermore, we see that number of bedrooms raises home value most for homes at the .25 quantile, where square footage does not necessarily have that effect. Therefore, if buyers are on a budget, maximizing value with a higher building area rather than looking for a home with more bedrooms may be recommended.

The last direction was to use property type as a parameter of interest to analyze the dataset. The LDA model was found to be a good tool to predict classification of property types: "h", and "u" with similar dataset. Even though the accuracy of the model is high, the model does not predict townhouses correctly, and therefore predicted classifications for townhome should be checked. This is because townhouses only consist of 78 of the 971 observations after data cleansing. For a more robust analysis, a bigger sample is required. CA indicates that if homebuyers are looking for properties in more affluent neighborhoods, they might have fewer selections of Property Types, but more selections in less affluent neighborhoods. Keeping all other variables constant, if a homebuyer prefers living in less affluent neighborhoods, they might have a higher chance of getting houses than in more affluent neighborhoods. Also, the model suggests price does play a big role in property type classifications. Therefore, a change in price will make a bigger impact on between-class classifications of property type than a change in other independent variables.

E. Appendix

1. Variable names in our data set

Sublevel: houses were grouped into 1-4, with lower numbers corresponding to being in more livable suburbs as per Melbourne's most livable suburbs ranking

Rooms: Number of rooms

Price: Price in Australian dollars

Method: S - property sold; SP - property sold prior; PI - property passed in; PN - sold prior not disclosed; SN - sold not disclosed; NB - no bid; VB - vendor bid; W - withdrawn prior to auction; SA - sold after auction; SS - sold after auction price not disclosed. N/A - price or highest bid not available.

TTHouse - residence type is house

THouse - residence type is townhouse

Tunit - residence type is unit or duplex

Date: Date sold

Distance: Distance from downtown Melbourne in Kilometres

Regionname: General Region (West, North West, North, North east ...etc)

Propertycount: Number of properties that exist in the suburb.

Bedroom2 : Scraped # of Bedrooms (from different source)

Bathroom: Number of Bathrooms

Car: Number of carspots

Landsize: Land Size in Metres

BuildingArea: Building Size in Metres

YearBuilt: Year the house was built

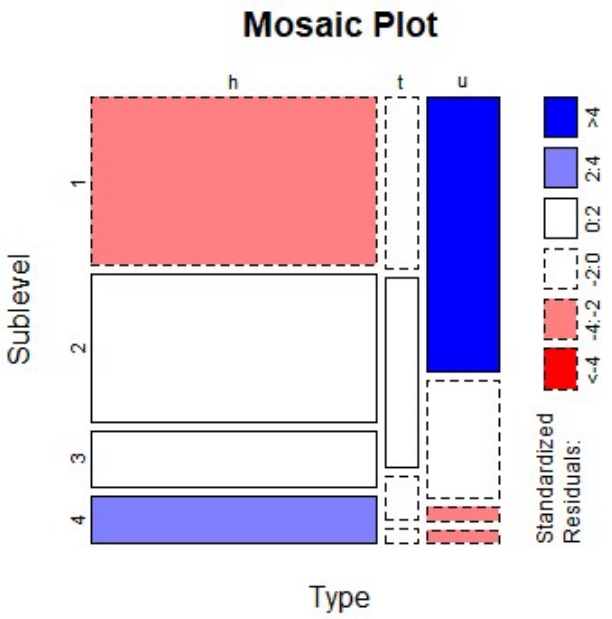
CouncilArea: Governing council for the area

Lattitude: Self explanatory

Longtitude: Self explanatory

2. CA and LDA

2.1 Mosaic Plot



2.2 Scale Table

Scaling	More Corresponsive	Less Corresponsive
"h"	Sublevel 2	Sublevel 1 & 4

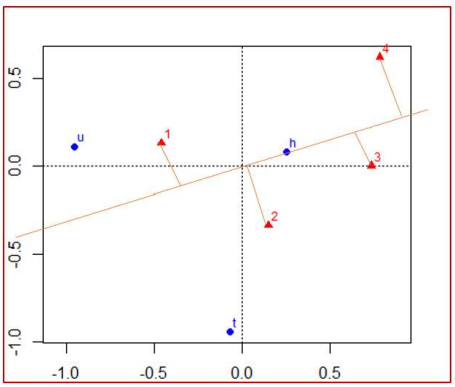
“t”	Sublevel 2	Sublevel 4
“u”	Sublevel 1	Sublevel 3

2.3 Chi-squared Test

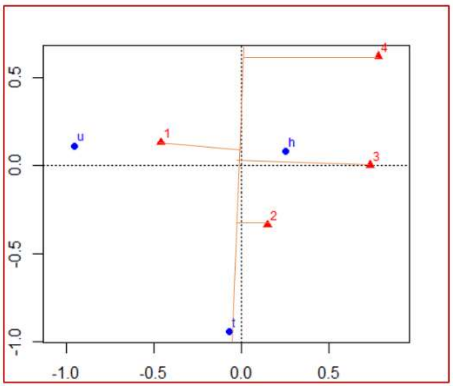
```
> chiSquared #singular value decomposition (SVD)
[1] 60.6432
> pchisq(chiSquared, (nrow(caData) - 1) * (ncol(caData) - 1), lower.tail=F)
[1] 3.331173e-11
```

2.4 Scaling

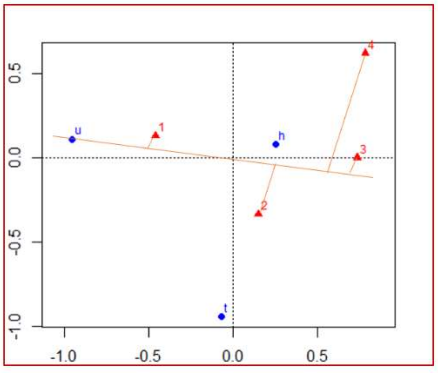
“h” scaling



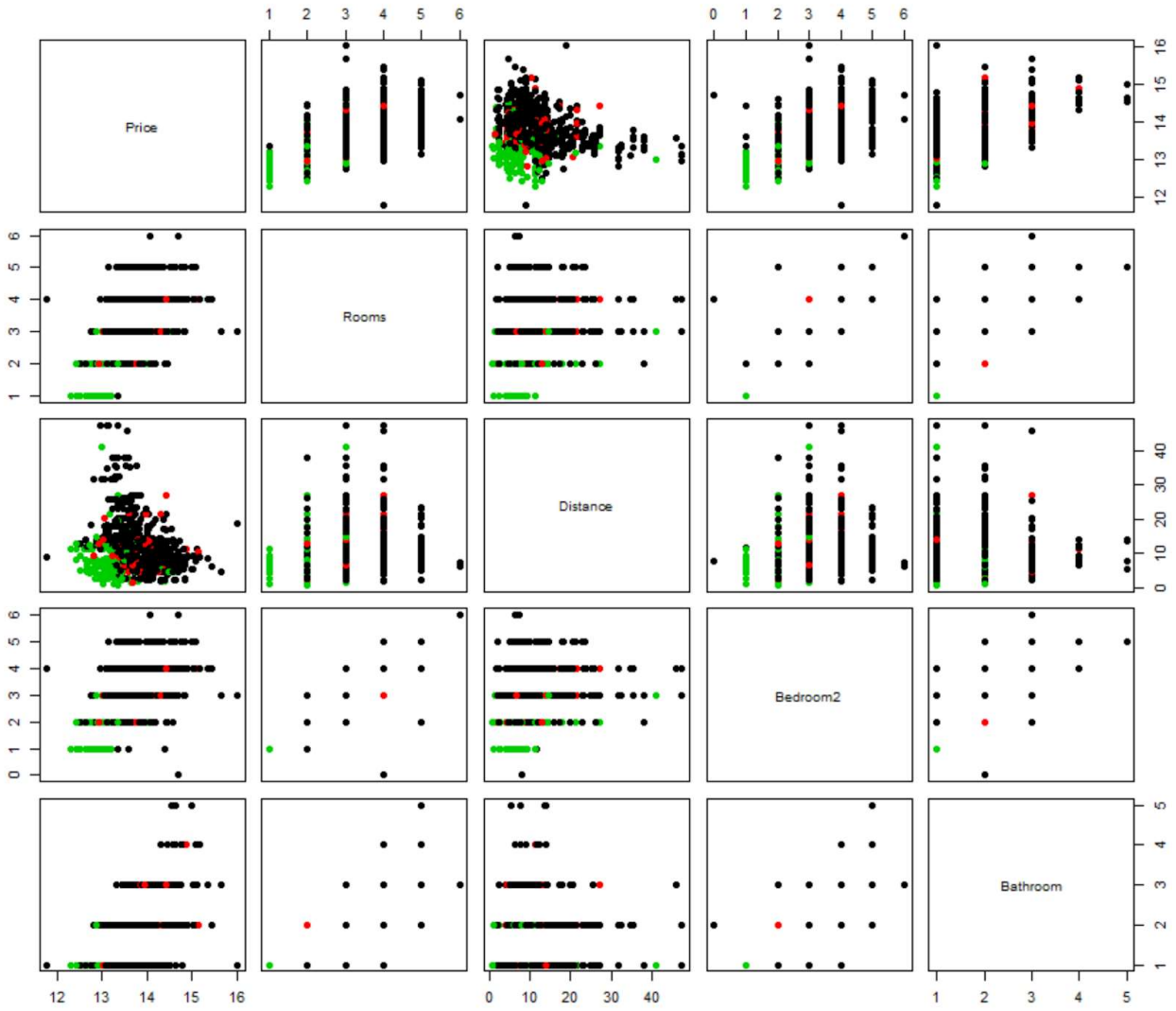
“t” scaling

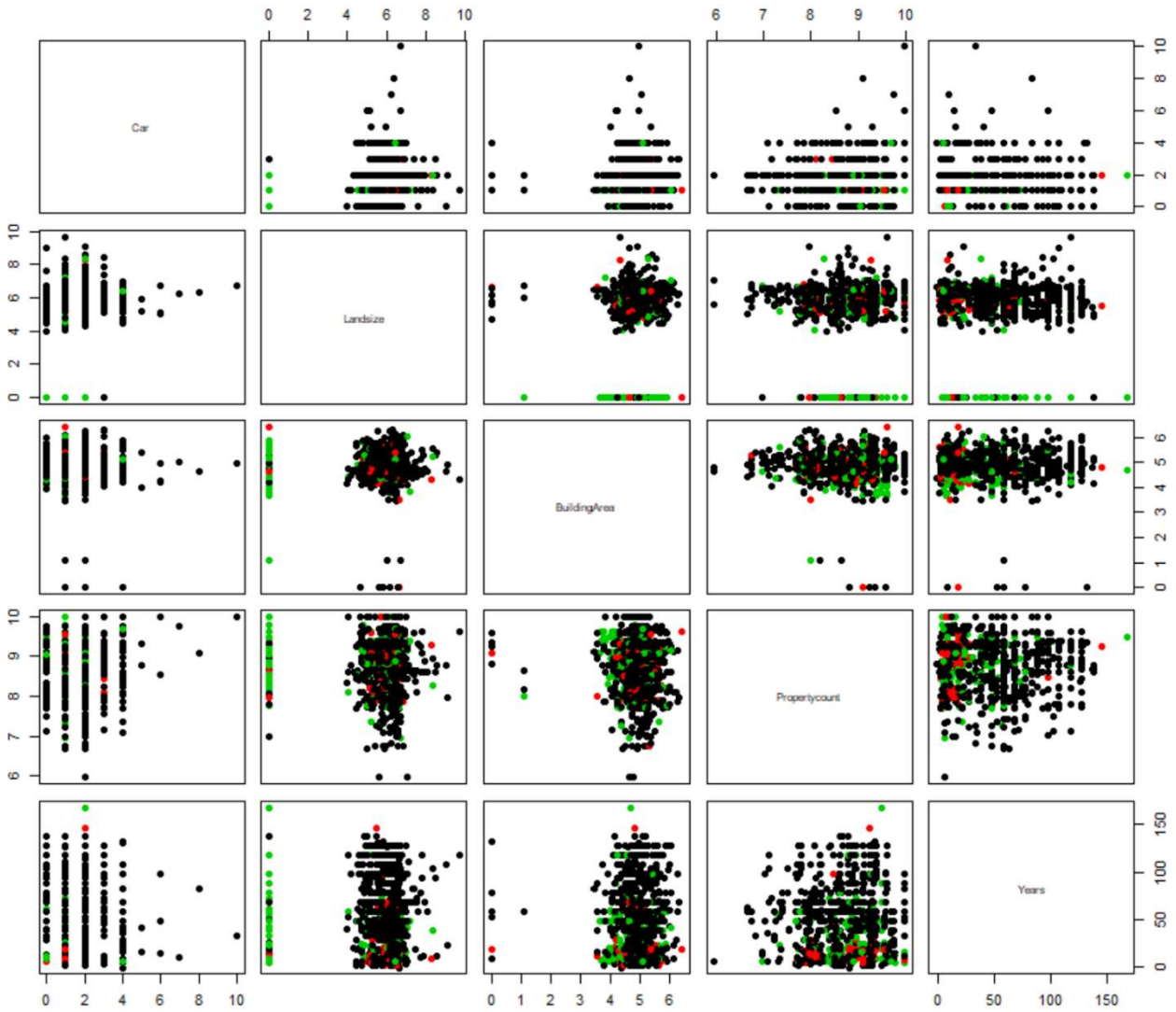


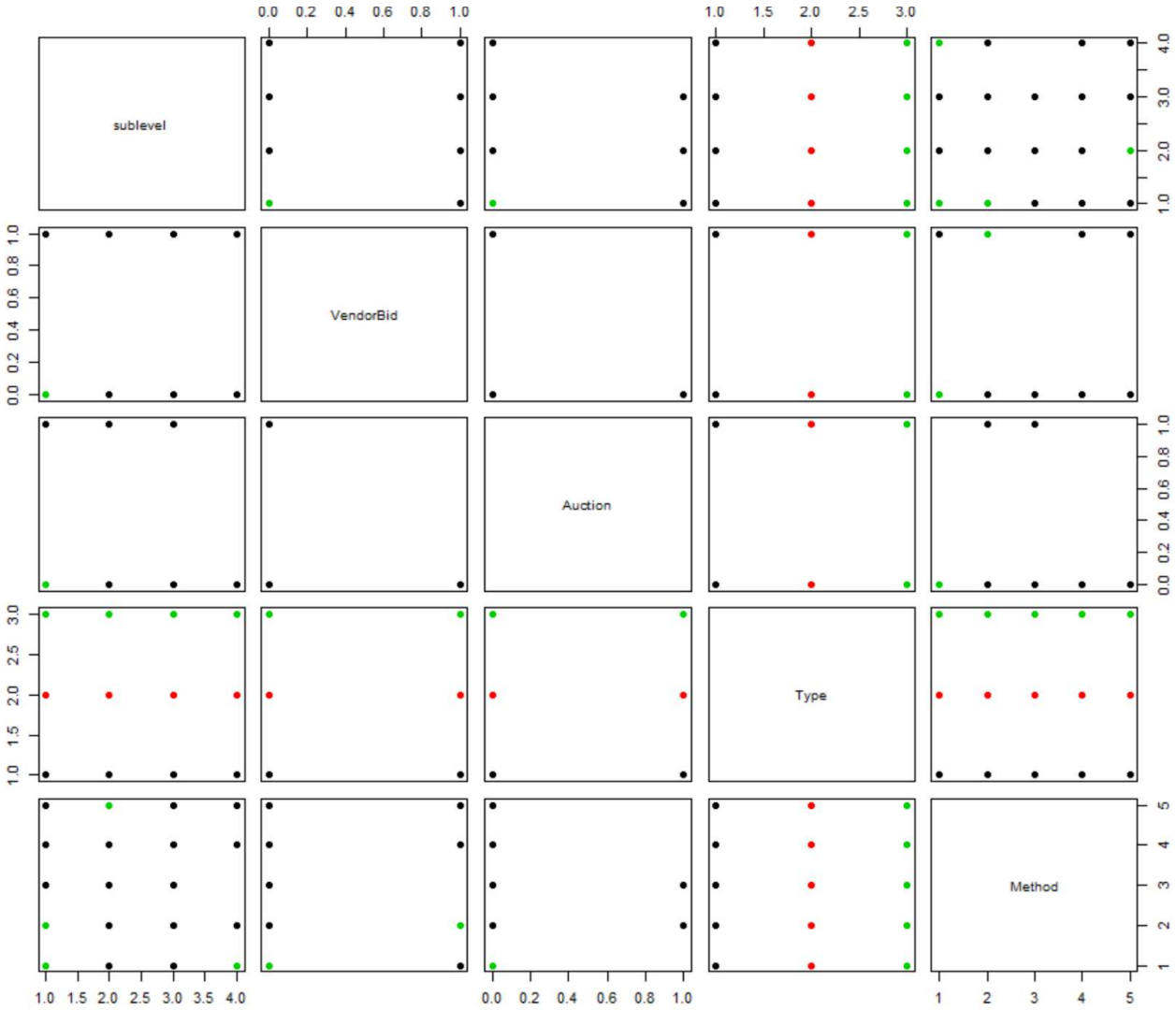
“u” scaling



2.5 Correlation Matrix


Price
Rooms
Distance
Bedroom2
Bathroom


Car
Landsize
BuildingArea
Propertycount
Years

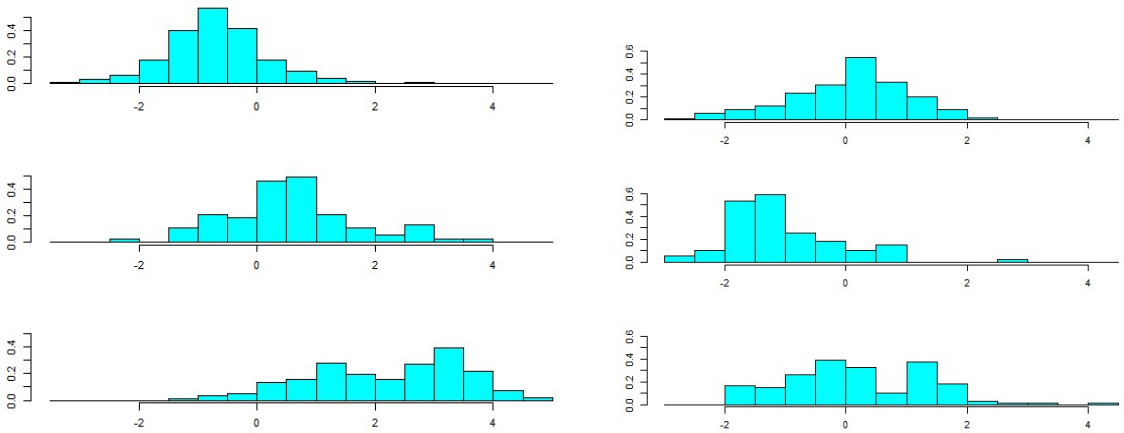

sublevel
VendorBid
Auction
Type
Method

2.6 Initial LDA results

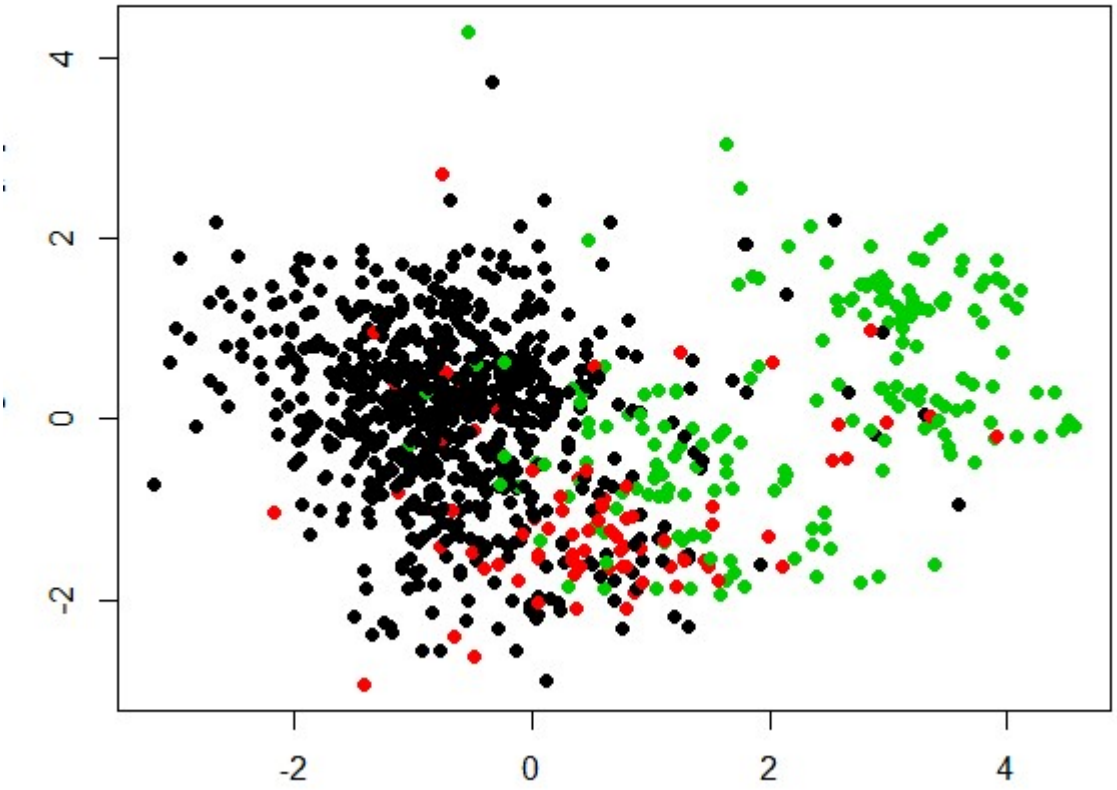
```
Call:
lda(Type ~ ., data = newHousing)

Prior probabilities of groups:
         h          t          u
0.72193615 0.08032956 0.19773429

Group means:
     Price    Rooms  Distance Bedroom2 Bathroom      Car Landsize BuildingArea Propertycount    Years
h 13.86956 3.336662 12.498573 3.315264 1.671897 1.807418 5.912354     4.885093      8.733695 58.92582
t 13.68962 2.974359 10.866667 2.923077 1.782051 1.576923 5.382949     4.810256      8.768205 20.48718
u 13.23234 1.994792  7.848438 1.984375 1.166667 1.125000 2.697344     4.656927      8.899323 36.35938
  sublevel  VendorBid     Auction   MethodS    MethodSA  MethodSP   MethodVB
h 2.024251 0.08273894 0.005706134 0.6619116 0.005706134 0.1283880 0.07988588
t 1.820513 0.11538462 0.012820513 0.6025641 0.012820513 0.1410256 0.12820513
u 1.437500 0.10937500 0.005208333 0.5729167 0.005208333 0.1718750 0.11458333

Coefficients of linear discriminants:
                      LD1          LD2
Price         -0.97111278 -0.522566004
Rooms         -0.40054797 -0.412725900
Distance      -0.02219115  0.023425165
Bedroom2      -0.15344626  0.540734574
Bathroom       0.09543785 -0.132821905
Car           -0.08465344  0.137006945
Landsize      -0.24658034 -0.262303725
BuildingArea   0.01263955  0.004011994
Propertycount  0.09746536 -0.033652307
Years         -0.01525446  0.029058261
sublevel      -0.43217428  0.143300294
VendorBid     -0.14228750  0.758041063
Auction        1.18026199 -0.297669885
MethodS        0.08585473 -0.071770615
MethodSA      -1.88231021 -0.809004560
MethodSP       0.25567582 -0.175960097
MethodVB       0.37335216 -0.903959738

Proportion of trace:
   LD1    LD2
0.9342 0.0658
```

2.7 Initial LDA Discriminant Histograms

LD1 LD2



2.8 Initial LDA Scatterplot



2.9 Initial LDA Performance Test

```
> table(newHousingLDA.values$class, newHousing$Type)

     h   t   u
  h 668  50  48
  t  15  15  14
  u  18  13 130
> confusion(newHousingLDA.values$class, newHousing$Type) #.8373
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.8373            0.7889            0.0453            0.1658

Confusion Matrix
      Predicted (cv)
Actual      h      t      u
     h 0.8721 0.0653 0.0627
     t 0.3409 0.3409 0.3182
     u 0.1118 0.0807 0.8075
```

2.10 LDA Performance Test on each Independent Variables

```
> newHousingLDAtrain1 = lda(Type~ Price, data=newHousingTrain)
> newHousingLDAtrain1.values = predict(newHousingLDAtrain1)
> confusion(newHousingLDAtrain1.values$class, newHousingTrain$Type) #.768
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.7680            0.8969            0.0000            0.1031

Confusion Matrix
      Predicted (cv)
Actual      h      t      u
     h 0.7730 0.0862 0.1408
     t    NaN    NaN    NaN
     u 0.2125 0.0625 0.7250

> newHousingLDAtrain2 = lda(Type~ Rooms, data=newHousingTrain)
> newHousingLDAtrain2.values = predict(newHousingLDAtrain2)
> confusion(newHousingLDAtrain2.values$class, newHousingTrain$Type) #.7809
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.7809            0.7139            0.0000            0.2861

Confusion Matrix
      Predicted (cv)
Actual      h      t      u
     h 0.8610 0.0903 0.0487
     t    NaN    NaN    NaN
     u 0.3514 0.0676 0.5811

> newHousingLDAtrain3 = lda(Type~ Distance, data=newHousingTrain)
> newHousingLDAtrain3.values = predict(newHousingLDAtrain3)
> confusion(newHousingLDAtrain3.values$class, newHousingTrain$Type) #.7152
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.7152            1.0000            0.0000            0.0000

Confusion Matrix
      Predicted (cv)
Actual      h      t     u
     h 0.7152 0.0838 0.201
     t    NaN    NaN   NaN
     u    NaN    NaN   NaN

> newHousingLDAtrain4 = lda(Type~ Bedroom2, data=newHousingTrain)
> newHousingLDAtrain4.values = predict(newHousingLDAtrain4)
> confusion(newHousingLDAtrain4.values$class, newHousingTrain$Type) #.7784
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.7784            0.7101            0.0000            0.2899

Confusion Matrix
      Predicted (cv)
Actual      h      t      u
     h 0.8621 0.0889 0.0490
     t    NaN    NaN    NaN
     u 0.3556 0.0711 0.5733
```

```
> newHousingLDAtrain4 = lda(Type~ Bedroom2, data=newHousingTrain)
> newHousingLDAtrain4.values = predict(newHousingLDAtrain4)
> confusion(newHousingLDAtrain4.values$class, newHousingTrain$Type) #.7784
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.7784            0.7101            0.0000            0.2899

Confusion Matrix
      Predicted (cv)
Actual      h      t      u
     h 0.8621 0.0889 0.0490
     t    NaN    NaN    NaN
     u 0.3556 0.0711 0.5733

> newHousingLDAtrain5 = lda(Type~ Bathroom, data=newHousingTrain)
> newHousingLDAtrain5.values = predict(newHousingLDAtrain5)
> confusion(newHousingLDAtrain5.values$class, newHousingTrain$Type) #.7152
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.7152            1.0000            0.0000            0.0000

Confusion Matrix
      Predicted (cv)
Actual      h      t     u
     h 0.7152 0.0838 0.201
     t    NaN    NaN   NaN
     u    NaN    NaN   NaN

> newHousingLDAtrain6 = lda(Type~ Car, data=newHousingTrain)
> newHousingLDAtrain6.values = predict(newHousingLDAtrain6)
> confusion(newHousingLDAtrain6.values$class, newHousingTrain$Type) #.7152
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.7152            1.0000            0.0000            0.0000

Confusion Matrix
      Predicted (cv)
Actual      h      t     u
     h 0.7152 0.0838 0.201
     t    NaN    NaN   NaN
     u    NaN    NaN   NaN

> newHousingLDAtrain7 = lda(Type~ Landsize, data=newHousingTrain)
> newHousingLDAtrain7.values = predict(newHousingLDAtrain7)
> confusion(newHousingLDAtrain7.values$class, newHousingTrain$Type) #.8093
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.8093            0.8711            0.0000            0.1289

Confusion Matrix
      Predicted (cv)
Actual      h      t      u
     h 0.8047 0.0888 0.1065
     t    NaN    NaN    NaN
     u 0.1100 0.0500 0.8400
```

```
> newHousingLDAtrain8 = lda(Type~ BuildingArea, data=newHousingTrain)
> newHousingLDAtrain8.values = predict(newHousingLDAtrain8)
> confusion(newHousingLDAtrain8.values$class, newHousingTrain$Type) #.7088
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.7088            0.9910            0.0000            0.0090

Confusion Matrix
      Predicted (cv)
Actual      h      t      u
     h 0.7139 0.0845 0.2016
     t    NaN    NaN    NaN
     u 0.8571 0.0000 0.1429

> newHousingLDAtrain9 = lda(Type~ Propertycount, data=newHousingTrain)
> newHousingLDAtrain9.values = predict(newHousingLDAtrain9)
> confusion(newHousingLDAtrain9.values$class, newHousingTrain$Type) #.7152
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.7152            1.0000            0.0000            0.0000

Confusion Matrix
      Predicted (cv)
Actual      h      t     u
     h 0.7152 0.0838 0.201
     t    NaN    NaN   NaN
     u    NaN    NaN   NaN

> newHousingLDAtrain10 = lda(Type~ Years, data=newHousingTrain)
> newHousingLDAtrain10.values = predict(newHousingLDAtrain10)
> confusion(newHousingLDAtrain10.values$class, newHousingTrain$Type) #.7152
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.7152            1.0000            0.0000            0.0000

Confusion Matrix
      Predicted (cv)
Actual      h      t     u
     h 0.7152 0.0838 0.201
     t    NaN    NaN   NaN
     u    NaN    NaN   NaN

> newHousingLDAtrain11 = lda(Type~ sublevel, data=newHousingTrain)
> newHousingLDAtrain11.values = predict(newHousingLDAtrain11)
> confusion(newHousingLDAtrain11.values$class, newHousingTrain$Type) #.7152
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.7152            1.0000            0.0000            0.0000

Confusion Matrix
      Predicted (cv)
Actual      h      t     u
     h 0.7152 0.0838 0.201
     t    NaN    NaN   NaN
     u    NaN    NaN   NaN
```

```
> newHousingLDAtrain12 = lda(Type~ VendorBid, data=newHousingTrain)
> newHousingLDAtrain12.values = predict(newHousingLDAtrain12)
> confusion(newHousingLDAtrain12.values$class, newHousingTrain$Type) #.7152
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.7152            1.0000            0.0000            0.0000

Confusion Matrix
      Predicted (cv)
Actual      h      t     u
     h 0.7152 0.0838 0.201
     t    NaN    NaN   NaN
     u    NaN    NaN   NaN

> newHousingLDAtrain13 = lda(Type~ Auction, data=newHousingTrain)
> newHousingLDAtrain13.values = predict(newHousingLDAtrain13)
> confusion(newHousingLDAtrain13.values$class, newHousingTrain$Type) #.7152
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.7152            1.0000            0.0000            0.0000

Confusion Matrix
      Predicted (cv)
Actual      h      t     u
     h 0.7152 0.0838 0.201
     t    NaN    NaN   NaN
     u    NaN    NaN   NaN

> newHousingLDAtrain14 = lda(Type~ Method, data=newHousingTrain)
> newHousingLDAtrain14.values = predict(newHousingLDAtrain14)
> confusion(newHousingLDAtrain14.values$class, newHousingTrain$Type) #.7139
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.7139            0.9948            0.0052            0.0000

Confusion Matrix
      Predicted (cv)
Actual      h      t      u
     h 0.7163 0.0829 0.2008
     t 0.5000 0.2500 0.2500
     u    NaN    NaN    NaN
```

2.11 LDA model built using test set and results

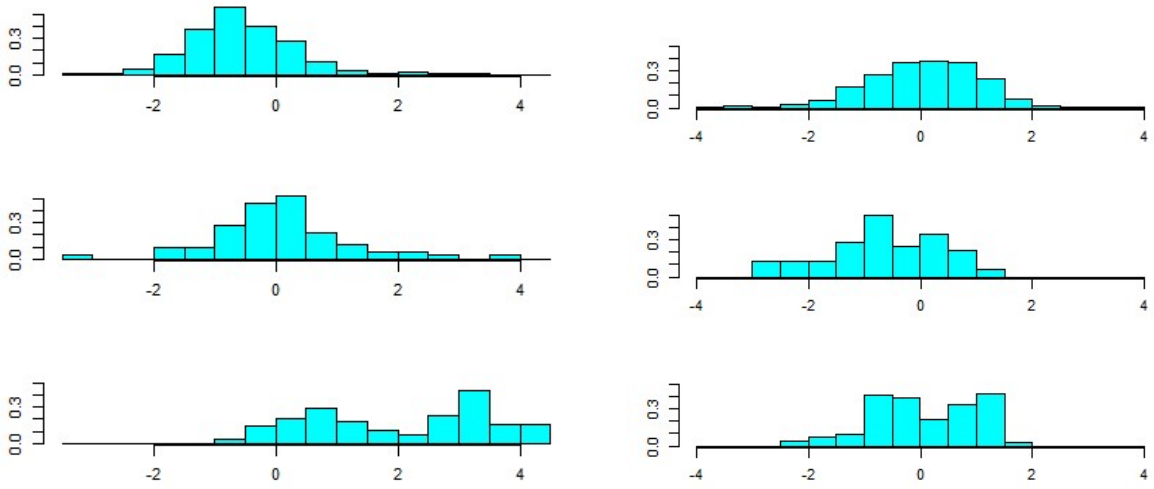
```
> newHousingLDAtrain15 = lda(Type~ Landsize+Rooms
+                            +Price+Distance+Bathroom, data=newHousingTrain)
> newHousingLDAtrain15
Call:
lda(Type ~ Landsize + Rooms + Price + Distance + Bathroom, data = newHousingTrain)

Prior probabilities of groups:
         h          t          u
0.71520619 0.08376289 0.20103093

Group means:
  Landsize    Rooms    Price  Distance Bathroom
h 5.906937 3.331532 13.86396 12.496036 1.659459
t 5.611692 2.953846 13.68985 11.172308 1.800000
u 2.763654 2.012821 13.22808  7.748077 1.166667

Coefficients of linear discriminants:
                LD1         LD2
Landsize -0.32196940 -0.32972071
Rooms    -0.67003407  0.88076474
Price    -1.15069304  0.89518071
Distance -0.04626747  0.02836725
Bathroom  0.44183317 -1.70627459

Proportion of trace:
   LD1    LD2
0.9691 0.0309
```

2.12 LDA full model performance test using test set and results

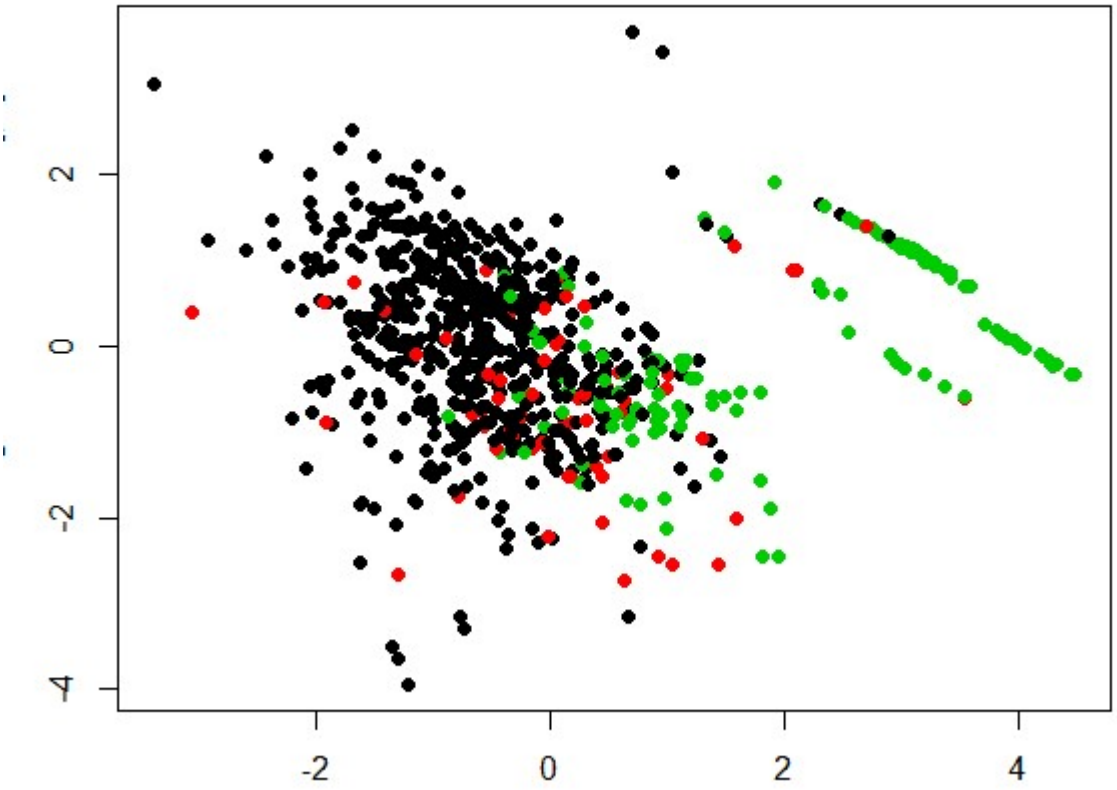
```
> table(newHousingLDAtrain15.values$class, newHousingTrain$Type)

     h   t   u
  h 542  54  60
  t   1   3   0
  u  12   8  96
> confusion(newHousingLDAtrain15.values$class, newHousingTrain$Type) #.826
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.8260            0.8454            0.0052            0.1495

Confusion Matrix
      Predicted (cv)
Actual      h      t      u
     h 0.8262 0.0823 0.0915
     t 0.2500 0.7500 0.0000
     u 0.1034 0.0690 0.8276
```

2.13 LDA final model discriminant histograms using train set

LD1 LD2



2.14 LDA full model scatterplot

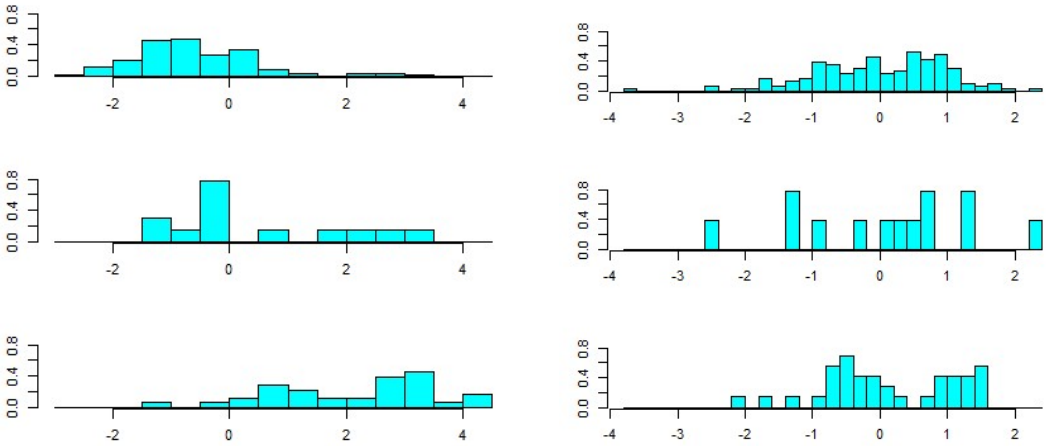


2.15 LDA final model performance using test set and results

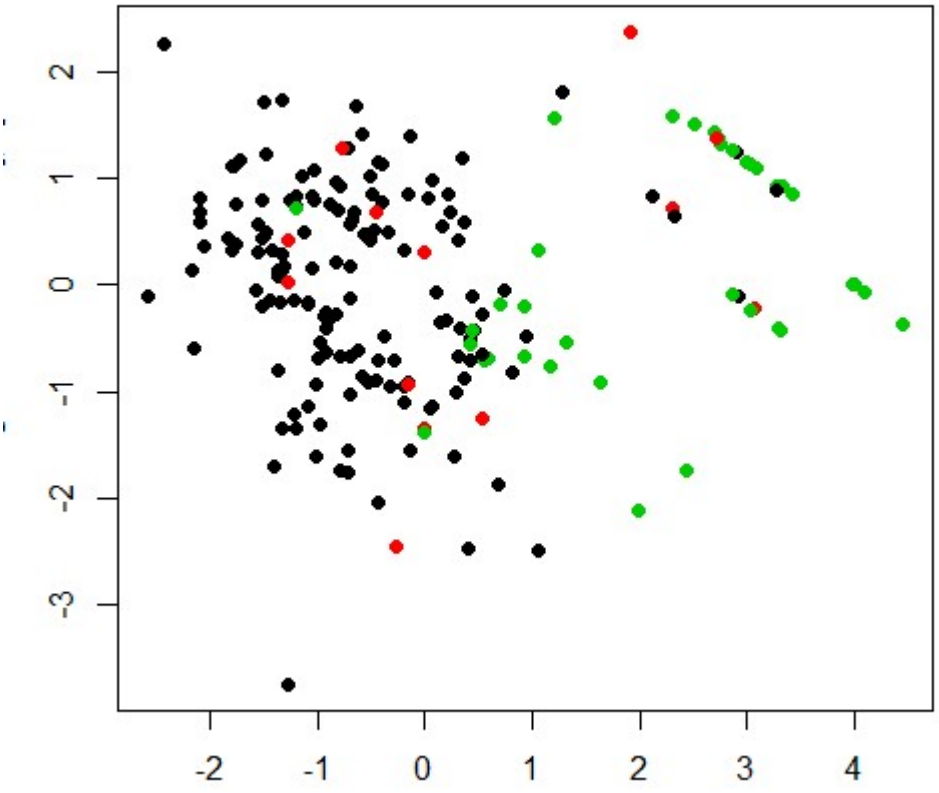
```
> table(newHousingLDAtest.values$class, newHousingTest$Type)

     h   t   u
  h 139   9  12
  t   1   0   0
  u   6   4  24
> confusion(newHousingLDAtest.values$class, newHousingTest$Type) #.8359
         Accuracy Prior Frequency.h Prior Frequency.t Prior Frequency.u
           0.8359            0.8205            0.0051            0.1744

Confusion Matrix
      Predicted (cv)
Actual      h      t      u
     h 0.8688 0.0562 0.0750
     t 1.0000 0.0000 0.0000
     u 0.1765 0.1176 0.7059
```

2.16 LDA final model discriminant histograms using train set

LD1 LD2



2.17 LDA final model scatterplot using train set



2.18 LDA final model prediction

```
> newHousingLDAtest.values
$class
  [1] h h h h h u h h h h h h h h h h h h h u h u h u h h h h u h h u h h h h u h h h h h u u u u h h u h h h h h
 [55] h h u u h h h h h u u u u u h h h h h u h u h h u u h u h u h h u h h h h u h h h u h h h h h h u h h h h h
[109] h t h h h h h h h h h h h u h h h h h u h h h h h h h h h h u h h h h h h h h h h h h h h h h h h u u h h h
[163] h h h h h h h h h h h h h h h h h h h h h h h h h h h h h h h h h
Levels: h t u

$posterior
              h            t            u
4   0.8718466488 0.1253987645 2.754587e-03
8   0.9614317463 0.0384310419 1.372119e-04
11  0.9013272054 0.0941387383 4.534056e-03
14  0.5388146896 0.1014037175 3.597816e-01
27  0.7650061165 0.1971526334 3.784125e-02
```