DFA

KEY

Wed Apr 15 14:57:38 2020

## Discriminant analysis

We will be using Discriminant Function Analysis (DFA) to help us understand the multivariate differences between shells, and to predict cover type from LandSat imagery data.

To begin, import the shells data:

library(readxl)  
read\_excel("multivariate\_data\_fixed.xlsx","shells") -> shells

Load the MASS library:

library(MASS)

Conduct the Linear Discriminant Analysis (lda, which is what the MASS package calls Discriminant Function Analysis):

lda(shell.type ~ major.axis + minor.axis + surface.len + height + depth + ln.weight, data = shells) -> shells.lda  
  
shells.lda

## Call:  
## lda(shell.type ~ major.axis + minor.axis + surface.len + height +   
## depth + ln.weight, data = shells)  
##   
## Prior probabilities of groups:  
## Calico White   
## 0.5 0.5   
##   
## Group means:  
## major.axis minor.axis surface.len height depth ln.weight  
## Calico 30.4775 23.91333 34.98833 7.089167 5.084167 1.0597323  
## White 27.9360 20.86150 34.77833 9.155333 7.529167 0.4377096  
##   
## Coefficients of linear discriminants:  
## LD1  
## major.axis -0.008631750  
## minor.axis -0.228773082  
## surface.len -0.002306157  
## height 2.736485922  
## depth 0.330688862  
## ln.weight -8.130160876

**Question: are all of the coefficients the same sign? What does this tell you about whether the shells are just different in size, or differ in shape?**

No, height and depth are positive and the rest are negative.

Use predict() to obtain LD1 scores, which we will use to calculate loadings:

predict(shells.lda) -> shells.pred  
  
shells.pred

## $class  
## [1] Calico Calico Calico Calico Calico Calico Calico Calico Calico Calico  
## [11] Calico Calico Calico Calico Calico Calico Calico Calico Calico Calico  
## [21] Calico Calico Calico Calico Calico Calico Calico Calico Calico Calico  
## [31] Calico Calico Calico Calico Calico Calico Calico Calico Calico Calico  
## [41] Calico Calico Calico Calico Calico Calico Calico Calico Calico Calico  
## [51] Calico Calico Calico Calico Calico Calico Calico Calico Calico Calico  
## [61] White White White White White White White White White White   
## [71] White White White White White White White White White White   
## [81] White White White White White White White White White White   
## [91] White White White White White White White White White White   
## [101] White White White White White White White White White White   
## [111] White White White White White White White White White White   
## Levels: Calico White  
##   
## $posterior  
## Calico White  
## 1 1.000000e+00 2.579304e-32  
## 2 1.000000e+00 3.156622e-32  
## 3 1.000000e+00 8.005436e-39  
## 4 1.000000e+00 1.154423e-33  
## 5 1.000000e+00 5.479819e-34  
## 6 1.000000e+00 7.596142e-33  
## 7 1.000000e+00 1.870099e-32  
## 8 1.000000e+00 2.896402e-25  
## 9 1.000000e+00 6.209634e-12  
## 10 1.000000e+00 5.218251e-27  
## 11 1.000000e+00 1.539908e-34  
## 12 1.000000e+00 9.902799e-24  
## 13 1.000000e+00 4.101382e-34  
## 14 1.000000e+00 1.774328e-28  
## 15 1.000000e+00 2.482507e-30  
## 16 1.000000e+00 9.697980e-28  
## 17 1.000000e+00 8.876186e-32  
## 18 1.000000e+00 1.044207e-44  
## 19 1.000000e+00 9.096996e-40  
## 20 1.000000e+00 2.629722e-39  
## 21 1.000000e+00 5.940894e-45  
## 22 1.000000e+00 1.651976e-40  
## 23 1.000000e+00 9.727599e-40  
## 24 1.000000e+00 2.224543e-33  
## 25 1.000000e+00 2.096215e-36  
## 26 1.000000e+00 8.243943e-31  
## 27 1.000000e+00 1.567975e-27  
## 28 1.000000e+00 2.533253e-37  
## 29 1.000000e+00 4.365885e-38  
## 30 1.000000e+00 2.715831e-38  
## 31 1.000000e+00 1.472369e-36  
## 32 1.000000e+00 2.506703e-41  
## 33 1.000000e+00 6.710556e-27  
## 34 1.000000e+00 6.135224e-36  
## 35 1.000000e+00 9.107920e-33  
## 36 1.000000e+00 3.771777e-37  
## 37 1.000000e+00 9.223939e-23  
## 38 1.000000e+00 1.508639e-31  
## 39 1.000000e+00 3.348284e-28  
## 40 1.000000e+00 1.951224e-26  
## 41 1.000000e+00 1.028549e-35  
## 42 1.000000e+00 2.504160e-30  
## 43 1.000000e+00 6.934154e-35  
## 44 1.000000e+00 2.019574e-29  
## 45 1.000000e+00 6.269328e-33  
## 46 1.000000e+00 8.496227e-31  
## 47 1.000000e+00 1.366782e-33  
## 48 1.000000e+00 6.480022e-33  
## 49 1.000000e+00 7.777941e-42  
## 50 1.000000e+00 3.117494e-35  
## 51 1.000000e+00 2.344415e-32  
## 52 1.000000e+00 2.814581e-35  
## 53 1.000000e+00 5.157012e-40  
## 54 1.000000e+00 2.885351e-35  
## 55 1.000000e+00 3.204115e-35  
## 56 1.000000e+00 1.587745e-27  
## 57 1.000000e+00 2.223599e-30  
## 58 1.000000e+00 1.090511e-32  
## 59 1.000000e+00 1.989402e-37  
## 60 1.000000e+00 3.363625e-30  
## 61 1.010296e-26 1.000000e+00  
## 62 5.464287e-29 1.000000e+00  
## 63 1.770075e-35 1.000000e+00  
## 64 1.448581e-35 1.000000e+00  
## 65 6.530445e-41 1.000000e+00  
## 66 4.649570e-30 1.000000e+00  
## 67 2.691366e-37 1.000000e+00  
## 68 4.557327e-27 1.000000e+00  
## 69 8.733134e-36 1.000000e+00  
## 70 9.106995e-37 1.000000e+00  
## 71 3.882592e-35 1.000000e+00  
## 72 1.834803e-36 1.000000e+00  
## 73 5.004161e-39 1.000000e+00  
## 74 3.774958e-33 1.000000e+00  
## 75 4.574643e-32 1.000000e+00  
## 76 2.814044e-35 1.000000e+00  
## 77 1.529428e-35 1.000000e+00  
## 78 9.744037e-31 1.000000e+00  
## 79 4.927994e-35 1.000000e+00  
## 80 3.820147e-27 1.000000e+00  
## 81 1.413755e-31 1.000000e+00  
## 82 6.473164e-31 1.000000e+00  
## 83 5.378705e-37 1.000000e+00  
## 84 1.831700e-33 1.000000e+00  
## 85 6.076589e-31 1.000000e+00  
## 86 4.233381e-33 1.000000e+00  
## 87 2.609115e-28 1.000000e+00  
## 88 4.328522e-36 1.000000e+00  
## 89 3.168853e-31 1.000000e+00  
## 90 2.557477e-32 1.000000e+00  
## 91 8.046698e-25 1.000000e+00  
## 92 1.161577e-23 1.000000e+00  
## 93 3.528157e-40 1.000000e+00  
## 94 2.593839e-37 1.000000e+00  
## 95 8.574875e-55 1.000000e+00  
## 96 1.759460e-29 1.000000e+00  
## 97 1.204937e-29 1.000000e+00  
## 98 5.298977e-27 1.000000e+00  
## 99 1.647327e-39 1.000000e+00  
## 100 3.666349e-38 1.000000e+00  
## 101 7.017609e-30 1.000000e+00  
## 102 5.919848e-37 1.000000e+00  
## 103 1.093525e-32 1.000000e+00  
## 104 2.823730e-34 1.000000e+00  
## 105 1.376783e-27 1.000000e+00  
## 106 2.480070e-31 1.000000e+00  
## 107 2.197510e-33 1.000000e+00  
## 108 5.689435e-35 1.000000e+00  
## 109 5.737845e-31 1.000000e+00  
## 110 4.082689e-23 1.000000e+00  
## 111 2.893884e-34 1.000000e+00  
## 112 2.920326e-34 1.000000e+00  
## 113 3.610044e-32 1.000000e+00  
## 114 6.405215e-34 1.000000e+00  
## 115 2.454667e-32 1.000000e+00  
## 116 1.684830e-30 1.000000e+00  
## 117 4.584825e-30 1.000000e+00  
## 118 3.168519e-38 1.000000e+00  
## 119 9.994799e-32 1.000000e+00  
## 120 2.756538e-39 1.000000e+00  
##   
## $x  
## LD1  
## 1 -5.942266  
## 2 -5.925764  
## 3 -7.166540  
## 4 -6.196059  
## 5 -6.256933  
## 6 -6.042138  
## 7 -5.968534  
## 8 -4.615989  
## 9 -2.108191  
## 10 -4.944124  
## 11 -6.360635  
## 12 -4.327440  
## 13 -6.280604  
## 14 -5.220369  
## 15 -5.569161  
## 16 -5.081607  
## 17 -5.841300  
## 18 -8.273520  
## 19 -7.344212  
## 20 -7.257488  
## 21 -8.319596  
## 22 -7.483585  
## 23 -7.338736  
## 24 -6.142469  
## 25 -6.711668  
## 26 -5.659222  
## 27 -5.042355  
## 28 -6.884312  
## 29 -7.027958  
## 30 -7.066741  
## 31 -6.740529  
## 32 -7.637632  
## 33 -4.923576  
## 34 -6.623933  
## 35 -6.027310  
## 36 -6.851793  
## 37 -4.145127  
## 38 -5.797966  
## 39 -5.168489  
## 40 -4.836375  
## 41 -6.581721  
## 42 -5.568452  
## 43 -6.425817  
## 44 -5.397907  
## 45 -6.057821  
## 46 -5.656760  
## 47 -6.182264  
## 48 -6.055121  
## 49 -7.733239  
## 50 -6.491128  
## 51 -5.950067  
## 52 -6.499479  
## 53 -7.390582  
## 54 -6.497450  
## 55 -6.488889  
## 56 -5.041332  
## 57 -5.578159  
## 58 -6.012597  
## 59 -6.904056  
## 60 -5.544346  
## 61 4.890150  
## 62 5.316590  
## 63 6.537369  
## 64 6.553745  
## 65 7.559407  
## 66 5.517896  
## 67 6.879366  
## 68 4.955188  
## 69 6.595087  
## 70 6.779777  
## 71 6.473198  
## 72 6.722550  
## 73 7.204925  
## 74 6.099265  
## 75 5.895453  
## 76 6.499494  
## 77 6.549308  
## 78 5.645564  
## 79 6.453719  
## 80 4.969603  
## 81 5.803273  
## 82 5.678978  
## 83 6.822799  
## 84 6.158344  
## 85 5.684143  
## 86 6.089901  
## 87 5.188868  
## 88 6.652430  
## 89 5.737334  
## 90 5.942960  
## 91 4.532511  
## 92 4.314405  
## 93 7.421593  
## 94 6.882381  
## 95 10.170764  
## 96 5.409172  
## 97 5.440100  
## 98 4.942870  
## 99 7.295700  
## 100 7.042224  
## 101 5.484265  
## 102 6.814967  
## 103 6.012371  
## 104 6.311099  
## 105 5.052979  
## 106 5.757356  
## 107 6.143468  
## 108 6.441981  
## 109 5.688829  
## 110 4.211714  
## 111 6.309094  
## 112 6.308351  
## 113 5.914799  
## 114 6.244184  
## 115 5.946312  
## 116 5.600827  
## 117 5.519041  
## 118 7.054146  
## 119 5.831603  
## 120 7.253641

Make a vector of response variable names:

shells.responses <- c("major.axis","minor.axis","surface.len","height","depth","ln.weight")

Calculate loadings (i.e. correlations between the data and the LD1 scores):

cor(shells[,shells.responses], shells.pred$x) -> shells.loadings  
  
shells.loadings

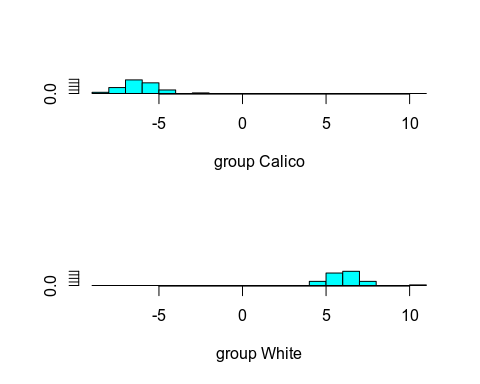
## LD1  
## major.axis -0.4222267  
## minor.axis -0.5642879  
## surface.len -0.0272162  
## height 0.7148903  
## depth 0.7523239  
## ln.weight -0.7053606

**Question: which variables are most strongly correlated with LD1? Are the signs the same for the loadings as for the coefficients?**

Yes, the signs are all the same (this won’t always be true, but they are the same for this data set).

Plot the histograms of scores for each shell.type group:

plot(shells.lda)



**Question: is there any overlap in shell morphology? How do yo know?**

There doesn’t appear to be - the histogram of LD1 scores for the Calico shells seems to end below 0 and the histogram for the White group starts well above zero.

**Question: how do white shells compare with calico shells morphologically, based on these results?**

Since white shells have big LD1 scores, they are taller and deeper than calico shells, but calico shells are heavier, have longer major and minor axes, and bigger suface lengths than white shells.

Obtain a confusion matrix that compares actual shell types to predicted shell types:

table(shells.pred$class, shells$shell.type, dnn = c("Predicted","Observed"))

## Observed  
## Predicted Calico White  
## Calico 60 0  
## White 0 60

**Question: do the white shells and calico shells separate completely, or is their overlap in their morphology? How can you tell (what would the confusion matrix look like if we couldn’t tell them apart)?**

As we expected from the histograms, the groups separate completely. All of the Calico shells are predicted to be Calico, and all the White shells are predicted to be White.

## LandSat Data - classifying cover types with reflectances

Import the data:

data.frame(read\_excel("multivariate\_data\_fixed.xlsx","landsat")) -> landsat

Transform all but band4 and band6

landsat$log.band1 <- log(landsat$band1)  
landsat$log.band2 <- log(landsat$band2)  
landsat$log.band3 <- log(landsat$band3)  
landsat$log.band5 <- log(landsat$band5)  
landsat$log.band7 <- log(landsat$band7)

Make a list of the variables:

landsat.variables <- c("log.band1","log.band2","log.band3","band4","log.band5","band6","log.band7")

Label rows with abbreviations of the cover types:

landsat$ct <- as.factor(abbreviate(landsat$cover.type))

Run the LDA:

lda(ct ~ log.band1 + log.band2 + log.band3 + band4 + log.band5 + band6 + log.band7, data=landsat) -> landsat.lda  
landsat.lda

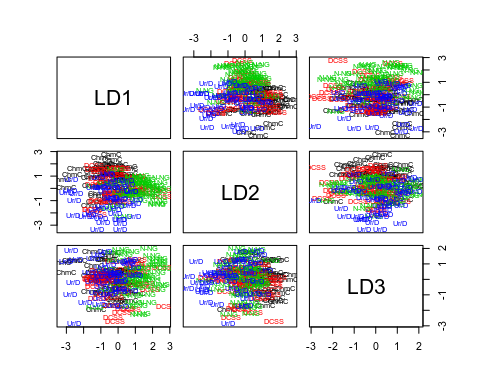
## Call:  
## lda(ct ~ log.band1 + log.band2 + log.band3 + band4 + log.band5 +   
## band6 + log.band7, data = landsat)  
##   
## Prior probabilities of groups:  
## ChmC DCSS N-NG Ur/D   
## 0.25 0.25 0.25 0.25   
##   
## Group means:  
## log.band1 log.band2 log.band3 band4 log.band5 band6 log.band7  
## ChmC 4.202917 3.456613 3.482140 67.22 4.407366 144.52 3.661511  
## DCSS 4.251808 3.529610 3.572739 71.50 4.472402 148.24 3.702492  
## N-NG 4.228638 3.497988 3.565184 71.94 4.520316 150.72 3.677978  
## Ur/D 4.338845 3.631407 3.696178 77.80 4.510531 150.70 3.791580  
##   
## Coefficients of linear discriminants:  
## LD1 LD2 LD3  
## log.band1 -5.64528569 -12.52305977 21.81166220  
## log.band2 1.85012837 4.88268785 -37.92582944  
## log.band3 5.38456123 -1.26311153 12.79056921  
## band4 -0.06877090 -0.04365170 0.08657798  
## log.band5 14.17160676 -0.03739565 -4.78362929  
## band6 0.05197806 -0.08812609 0.00115943  
## log.band7 -13.94435573 2.53912445 3.55563452  
##   
## Proportion of trace:  
## LD1 LD2 LD3   
## 0.5508 0.4291 0.0201

**Question: why are there three LD axes? Do we need all three to tell the cover types apart, based on the variance explained?**

You will get on less LD axes than there are groups - with four groups you get three axes.

Plot the scatterplot matrix of scores:

plot(landsat.lda, col = as.integer(landsat$ct))



**Question: does it look like the cover types separate completely on any of the LD axes? How can you tell?**

No, there is a lot of overlap in the colors that indicate different cover types.

Predict the cover types:

predict(landsat.lda) -> landsat.pred  
  
landsat.pred

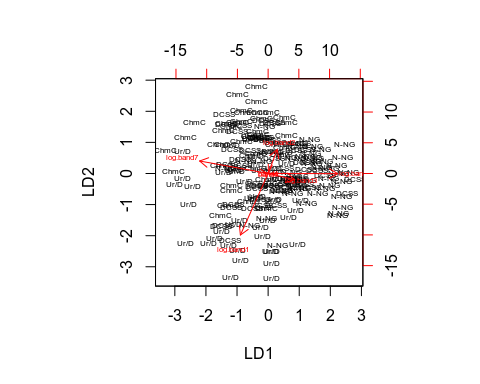
## $class  
## [1] ChmC N-NG N-NG ChmC N-NG ChmC DCSS DCSS ChmC ChmC ChmC ChmC DCSS ChmC  
## [15] DCSS DCSS ChmC ChmC DCSS ChmC N-NG ChmC ChmC ChmC ChmC ChmC ChmC ChmC  
## [29] Ur/D ChmC ChmC DCSS N-NG Ur/D Ur/D ChmC Ur/D ChmC ChmC ChmC DCSS ChmC  
## [43] Ur/D ChmC ChmC ChmC Ur/D ChmC ChmC ChmC Ur/D N-NG N-NG ChmC ChmC N-NG  
## [57] N-NG ChmC DCSS N-NG ChmC Ur/D DCSS DCSS N-NG ChmC N-NG ChmC Ur/D ChmC  
## [71] N-NG Ur/D ChmC Ur/D DCSS ChmC DCSS N-NG DCSS N-NG N-NG N-NG ChmC ChmC  
## [85] Ur/D DCSS DCSS ChmC N-NG ChmC Ur/D N-NG Ur/D Ur/D DCSS DCSS ChmC DCSS  
## [99] DCSS ChmC N-NG DCSS DCSS N-NG N-NG N-NG N-NG N-NG N-NG DCSS ChmC N-NG  
## [113] N-NG N-NG N-NG N-NG ChmC N-NG ChmC N-NG N-NG N-NG N-NG N-NG Ur/D N-NG  
## [127] N-NG N-NG N-NG N-NG N-NG Ur/D N-NG N-NG N-NG N-NG N-NG Ur/D DCSS N-NG  
## [141] DCSS N-NG N-NG ChmC N-NG N-NG N-NG N-NG N-NG DCSS Ur/D Ur/D Ur/D Ur/D  
## [155] Ur/D N-NG ChmC Ur/D Ur/D Ur/D Ur/D ChmC Ur/D Ur/D DCSS ChmC N-NG Ur/D  
## [169] N-NG Ur/D Ur/D ChmC Ur/D Ur/D ChmC Ur/D Ur/D ChmC Ur/D N-NG ChmC Ur/D  
## [183] Ur/D Ur/D Ur/D DCSS DCSS N-NG ChmC Ur/D Ur/D Ur/D DCSS ChmC Ur/D Ur/D  
## [197] DCSS Ur/D DCSS Ur/D  
## Levels: ChmC DCSS N-NG Ur/D  
##   
## $posterior  
## ChmC DCSS N-NG Ur/D  
## 1 0.866756514 0.11206723 0.0122795827 0.008896674  
## 2 0.096648685 0.35606238 0.3783162965 0.168972639  
## 3 0.312115585 0.30977904 0.3160267185 0.062078653  
## 4 0.737930866 0.18410399 0.0147314724 0.063233673  
## 5 0.104484963 0.23281096 0.5413231679 0.121380907  
## 6 0.753596954 0.18258054 0.0245992483 0.039223253  
## 7 0.360029838 0.36605070 0.2121236939 0.061795766  
## 8 0.328670521 0.38809185 0.2494711260 0.033766499  
## 9 0.798030168 0.13125191 0.0211390610 0.049578862  
## 10 0.515455313 0.26030896 0.2061897358 0.018045993  
## 11 0.750918833 0.19988637 0.0380881097 0.011106687  
## 12 0.823485557 0.12407974 0.0303952527 0.022039448  
## 13 0.203272652 0.33335785 0.1936092353 0.269760261  
## 14 0.792365745 0.14945991 0.0148725794 0.043301769  
## 15 0.413646644 0.42377581 0.1058415403 0.056736001  
## 16 0.266842848 0.37665590 0.2754899582 0.081011296  
## 17 0.732000750 0.19299134 0.0545774147 0.020430499  
## 18 0.421876259 0.30119936 0.2125692976 0.064355084  
## 19 0.383431099 0.39843941 0.1991667218 0.018962773  
## 20 0.367201943 0.24364024 0.1357587928 0.253399028  
## 21 0.086466088 0.37794205 0.4409987407 0.094593122  
## 22 0.609127038 0.25452901 0.0142607213 0.122083235  
## 23 0.729026698 0.20510964 0.0299801394 0.035883517  
## 24 0.561056899 0.33940835 0.0827792555 0.016755491  
## 25 0.513754379 0.15798879 0.0245594217 0.303697413  
## 26 0.540292780 0.29272211 0.0985901831 0.068394929  
## 27 0.726184876 0.06646978 0.0009833987 0.206361942  
## 28 0.850892115 0.11629872 0.0274107626 0.005398405  
## 29 0.230213124 0.27011297 0.0375779656 0.462095936  
## 30 0.884878642 0.06647116 0.0026157727 0.046034429  
## 31 0.457490306 0.10326596 0.0018147922 0.437428939  
## 32 0.191395415 0.42382616 0.2268153990 0.157963021  
## 33 0.264604725 0.29323514 0.3998322708 0.042327867  
## 34 0.127363328 0.19391241 0.0838109314 0.594913326  
## 35 0.072748420 0.23680722 0.1676343357 0.522810020  
## 36 0.435139095 0.33267823 0.1000893938 0.132093283  
## 37 0.085581168 0.29394385 0.0217293979 0.598745588  
## 38 0.547941988 0.32720335 0.0764500255 0.048404640  
## 39 0.819626483 0.07571351 0.0022309336 0.102429068  
## 40 0.588077708 0.32105171 0.0665771419 0.024293438  
## 41 0.337390123 0.38241330 0.1926612376 0.087535335  
## 42 0.754558680 0.20458877 0.0213989238 0.019453628  
## 43 0.067737177 0.12187472 0.0142830126 0.796105095  
## 44 0.564221279 0.30434460 0.0676903468 0.063743771  
## 45 0.434755555 0.43408123 0.0761462196 0.055016997  
## 46 0.505792862 0.23197084 0.0939668331 0.168269461  
## 47 0.204756743 0.21831500 0.0375367642 0.539391489  
## 48 0.676799964 0.17931339 0.0219118233 0.121974826  
## 49 0.541317877 0.33066952 0.0400680946 0.087944508  
## 50 0.594610119 0.25653374 0.0902938309 0.058562307  
## 51 0.085201979 0.21601545 0.0864564483 0.612326121  
## 52 0.224632760 0.26338389 0.4607016988 0.051281653  
## 53 0.119613176 0.35699782 0.3755215989 0.147867409  
## 54 0.510843269 0.31821731 0.1101333572 0.060806060  
## 55 0.830670514 0.13156559 0.0101121295 0.027651764  
## 56 0.072388666 0.20347948 0.6069122335 0.117219625  
## 57 0.031158534 0.20775041 0.7294537991 0.031637262  
## 58 0.498222210 0.34438554 0.0273559238 0.130036325  
## 59 0.146746019 0.38443590 0.1842990921 0.284518990  
## 60 0.117606833 0.27520260 0.5515247892 0.055665780  
## 61 0.457394041 0.28776791 0.0541013840 0.200736667  
## 62 0.112449327 0.18730479 0.0352783303 0.664967551  
## 63 0.076528574 0.47736451 0.2428806019 0.203226318  
## 64 0.454165939 0.46111138 0.0623452841 0.022377395  
## 65 0.006752666 0.06630576 0.9092579935 0.017683585  
## 66 0.375949134 0.32765919 0.0561423759 0.240249304  
## 67 0.235176645 0.24643645 0.4822988131 0.036088096  
## 68 0.697726136 0.21390177 0.0290193492 0.059352744  
## 69 0.166205067 0.32890847 0.1253314644 0.379554998  
## 70 0.409295840 0.26003973 0.0463804816 0.284283950  
## 71 0.083616299 0.29901173 0.5567062145 0.060665759  
## 72 0.060974630 0.34995883 0.2168838339 0.372182709  
## 73 0.578219369 0.27203631 0.0766991281 0.073045192  
## 74 0.039958345 0.09991157 0.0117471545 0.848382927  
## 75 0.074798145 0.47657618 0.0922683541 0.356357324  
## 76 0.760010217 0.18042763 0.0139839420 0.045578216  
## 77 0.391243040 0.40558852 0.1461116825 0.057056760  
## 78 0.047986580 0.19682986 0.6643645754 0.090818980  
## 79 0.205337218 0.54470577 0.1650005362 0.084956478  
## 80 0.105709767 0.29322906 0.3846856973 0.216375478  
## 81 0.006096740 0.11437320 0.8719388085 0.007591247  
## 82 0.095505290 0.27768352 0.5047842818 0.122026910  
## 83 0.418416251 0.29883364 0.0271753699 0.255574743  
## 84 0.566592128 0.28446626 0.1195295025 0.029412109  
## 85 0.018161231 0.07657008 0.0140003808 0.891268304  
## 86 0.085414778 0.48688232 0.3172887584 0.110414146  
## 87 0.196169253 0.35158157 0.1255675348 0.326681646  
## 88 0.522884306 0.31547283 0.1364321245 0.025210737  
## 89 0.054844934 0.23704212 0.6681728710 0.039940071  
## 90 0.497577161 0.29748387 0.0794787898 0.125460180  
## 91 0.039928102 0.10814832 0.0136955149 0.838228059  
## 92 0.161912244 0.26161188 0.5406021495 0.035873731  
## 93 0.095034283 0.19008955 0.0293100582 0.685566108  
## 94 0.092269664 0.21193178 0.1522181284 0.543580429  
## 95 0.187215436 0.34880245 0.3328060522 0.131176059  
## 96 0.143038400 0.33869998 0.2767430873 0.241518529  
## 97 0.453370610 0.39940342 0.0912430868 0.055982882  
## 98 0.167462313 0.35357818 0.1531341597 0.325825343  
## 99 0.142501537 0.46281440 0.0955026112 0.299181456  
## 100 0.653996809 0.27361057 0.0300813287 0.042311294  
## 101 0.052820283 0.15734864 0.7582003541 0.031630722  
## 102 0.064700331 0.41397770 0.4014249385 0.119897035  
## 103 0.100062361 0.32997226 0.3281149124 0.241850464  
## 104 0.066983300 0.23974716 0.5555883357 0.137681207  
## 105 0.223814959 0.33607116 0.4051750808 0.034938798  
## 106 0.030759311 0.12316953 0.8411799097 0.004891253  
## 107 0.163131246 0.22612987 0.4824415822 0.128297305  
## 108 0.011676795 0.07149684 0.9076659376 0.009160422  
## 109 0.020120939 0.14358594 0.8023468838 0.033946234  
## 110 0.149590402 0.33374427 0.3273074428 0.189357889  
## 111 0.444520367 0.28487058 0.0736215542 0.196987503  
## 112 0.095167000 0.28964244 0.5812845436 0.033906014  
## 113 0.027297920 0.18322009 0.7138714911 0.075610503  
## 114 0.005589573 0.06732960 0.8940359185 0.033044906  
## 115 0.064887520 0.40907286 0.4918803401 0.034159280  
## 116 0.095288448 0.20785832 0.6616370508 0.035216181  
## 117 0.416057325 0.23519078 0.2434018131 0.105350078  
## 118 0.175104489 0.24626052 0.5567728855 0.021862106  
## 119 0.643269951 0.20877660 0.1092907856 0.038662668  
## 120 0.076794826 0.39210570 0.4096941626 0.121405310  
## 121 0.116302438 0.25402371 0.5975583387 0.032115517  
## 122 0.181217379 0.27970409 0.4719653910 0.067113142  
## 123 0.027970554 0.10171750 0.8396358002 0.030676147  
## 124 0.025741364 0.09686450 0.8269381424 0.050455991  
## 125 0.045016137 0.24933390 0.1362235124 0.569426454  
## 126 0.024276082 0.12906494 0.8385837556 0.008075221  
## 127 0.076287251 0.27018536 0.6361067256 0.017420660  
## 128 0.064313942 0.33214051 0.4658336361 0.137711915  
## 129 0.034428488 0.16727925 0.7759579607 0.022334306  
## 130 0.027347611 0.09032717 0.8685819196 0.013743301  
## 131 0.013505497 0.07271925 0.8968322398 0.016943015  
## 132 0.035216365 0.19292332 0.0524254984 0.719434815  
## 133 0.091781722 0.27403502 0.5930891715 0.041094090  
## 134 0.129510817 0.25591902 0.5910876155 0.023482543  
## 135 0.131349859 0.27687497 0.5494521047 0.042323068  
## 136 0.121372055 0.23408139 0.3415369147 0.303009645  
## 137 0.008524618 0.07151870 0.8936841410 0.026272545  
## 138 0.011508271 0.14629523 0.1503370986 0.691859397  
## 139 0.320705631 0.39256232 0.0227852971 0.263946747  
## 140 0.024473958 0.38999442 0.4539227296 0.131608889  
## 141 0.169403827 0.49314770 0.2705832534 0.066865218  
## 142 0.053337066 0.38685422 0.4831364139 0.076672297  
## 143 0.005676484 0.07605470 0.8658081434 0.052460673  
## 144 0.408181209 0.24774920 0.1638892448 0.180180349  
## 145 0.017583962 0.14312666 0.8238315914 0.015457785  
## 146 0.048615231 0.28815375 0.6407102314 0.022520785  
## 147 0.068156597 0.30362180 0.4315115348 0.196710072  
## 148 0.209259266 0.32855363 0.4215578018 0.040629301  
## 149 0.005884627 0.08829794 0.8577835867 0.048033845  
## 150 0.125202995 0.42785236 0.2058105701 0.241134079  
## 151 0.079819899 0.24616442 0.1462572349 0.527758444  
## 152 0.008516689 0.13442928 0.0900905753 0.766963454  
## 153 0.036346605 0.18501093 0.3691925179 0.409449943  
## 154 0.113242059 0.33649330 0.1012637235 0.449000914  
## 155 0.002387761 0.02489527 0.0057233135 0.966993654  
## 156 0.111255398 0.24231417 0.4855852427 0.160845187  
## 157 0.377214016 0.25330830 0.0512806805 0.318197006  
## 158 0.020992634 0.14092951 0.0870643550 0.751013498  
## 159 0.043527255 0.20406650 0.1461987789 0.606207466  
## 160 0.015357326 0.03328365 0.0010957213 0.950263297  
## 161 0.028942199 0.09091677 0.0107496182 0.869391410  
## 162 0.351438345 0.32900098 0.1164663841 0.203094288  
## 163 0.014253702 0.07428623 0.0037831295 0.907676941  
## 164 0.085092910 0.31851798 0.2107245845 0.385664528  
## 165 0.096495417 0.38525714 0.3816685816 0.136578856  
## 166 0.383383824 0.27325151 0.0479786987 0.295385963  
## 167 0.292529746 0.28393259 0.3455768799 0.077960779  
## 168 0.051846088 0.15401961 0.0374432256 0.756691080  
## 169 0.216261963 0.28879676 0.3881273911 0.106813883  
## 170 0.005260548 0.06272499 0.0841225400 0.847891919  
## 171 0.303651810 0.07679696 0.0019126598 0.617638568  
## 172 0.412235661 0.18400314 0.0282782236 0.375482977  
## 173 0.032518333 0.16312402 0.0972222894 0.707135357  
## 174 0.005900767 0.05649544 0.0153167086 0.922287083  
## 175 0.375377458 0.37008243 0.1531120075 0.101428105  
## 176 0.013582663 0.08817173 0.0115728664 0.886672738  
## 177 0.291323464 0.20846596 0.0852053328 0.415005239  
## 178 0.716098520 0.07348535 0.0026550707 0.207761054  
## 179 0.328367895 0.17019096 0.0026968677 0.498744275  
## 180 0.042290847 0.24531850 0.5291653203 0.183225333  
## 181 0.675547728 0.16481668 0.0315461388 0.128089451  
## 182 0.008622358 0.14255002 0.0976501146 0.751177504  
## 183 0.037381096 0.20005454 0.0218096615 0.740754701  
## 184 0.009357685 0.14712199 0.3491217136 0.494398615  
## 185 0.280510183 0.10111147 0.0051071504 0.613271200  
## 186 0.229638905 0.38139966 0.2325960285 0.156365407  
## 187 0.211861133 0.49732467 0.1068179039 0.183996289  
## 188 0.037175243 0.20995570 0.4721027073 0.280766348  
## 189 0.631632299 0.23917384 0.0602582049 0.068935659  
## 190 0.097194961 0.20390835 0.0025449269 0.696351763  
## 191 0.133157231 0.22652052 0.1016924979 0.538629751  
## 192 0.125281080 0.29840608 0.1211574559 0.455155388  
## 193 0.314569631 0.37806569 0.0353547883 0.272009887  
## 194 0.511087633 0.35972368 0.0588909170 0.070297765  
## 195 0.010383258 0.07906532 0.0163596606 0.894191759  
## 196 0.091746167 0.30001726 0.2358676820 0.372368888  
## 197 0.129675298 0.50471333 0.1931705579 0.172440817  
## 198 0.020625516 0.05576755 0.0064806683 0.917126269  
## 199 0.355692730 0.41822253 0.0923130168 0.133771721  
## 200 0.002127781 0.06507926 0.0540815224 0.878711434  
##   
## $x  
## LD1 LD2 LD3  
## 1 -0.992161993 2.53907767 -0.034485142  
## 2 0.665641603 -0.35680948 -0.313267194  
## 3 0.540240468 0.83741322 0.589542393  
## 4 -1.423895134 1.37151180 -0.034783718  
## 5 0.938784962 -0.16689578 1.235407799  
## 6 -1.000498326 1.63251573 0.196696636  
## 7 0.280984744 0.94144401 -0.189838003  
## 8 0.577111554 1.23001197 -0.786402594  
## 9 -1.174337352 1.51739549 1.316955177  
## 10 0.529181425 1.80851542 0.315088253  
## 11 -0.378469229 2.33021672 -0.611755110  
## 12 -0.739029123 1.97408281 1.285067420  
## 13 -0.052924545 -0.19860304 0.456968634  
## 14 -1.327703957 1.61023928 0.414589542  
## 15 -0.119500345 1.09837197 -1.296186450  
## 16 0.429195252 0.61968820 -0.152987876  
## 17 -0.351569167 1.95829221 0.337871793  
## 18 0.224597789 0.99243288 0.648975493  
## 19 0.580620548 1.64761860 -1.377333179  
## 20 -0.398417077 0.14984705 1.733259646  
## 21 0.955144991 -0.09237520 -0.929030480  
## 22 -1.581693777 0.91826060 -0.798076759  
## 23 -0.852886242 1.66447154 -0.062578588  
## 24 0.019651798 1.94493682 -1.456074465  
## 25 -1.506696881 0.26535966 1.786763120  
## 26 -0.291933558 1.11781902 0.258944680  
## 27 -3.298554455 0.73470429 1.115761196  
## 28 -0.388313869 2.78333744 0.304352923  
## 29 -1.167727035 -0.38741907 -0.087757446  
## 30 -2.358513472 1.65016089 1.238317599  
## 31 -3.047466729 0.06663045 0.427804028  
## 32 0.213956508 0.07867131 -0.616868665  
## 33 0.830482876 0.95204229 0.508573270  
## 34 -0.635046449 -0.89782514 1.280905952  
## 35 -0.052841314 -1.14049918 0.613604759  
## 36 -0.418566394 0.63768754 0.127744660  
## 37 -1.279870973 -1.04215586 -1.969511186  
## 38 -0.334267125 1.33582856 -0.619820047  
## 39 -2.663097121 1.17225491 1.196355089  
## 40 -0.226301670 1.76583704 -1.167511496  
## 41 0.141785120 0.71653411 -0.236218999  
## 42 -0.869191731 2.03969549 -0.861550390  
## 43 -1.545169352 -1.36587791 0.253553571  
## 44 -0.493087639 1.19680037 -0.272497362  
## 45 -0.308185303 1.15592046 -1.691103514  
## 46 -0.570981380 0.56343686 1.560856697  
## 47 -1.184561706 -0.55033235 0.551291100  
## 48 -1.373430900 0.93913618 0.889984272  
## 49 -0.870296802 1.01728926 -0.897611237  
## 50 -0.322674172 1.25217619 0.571103206  
## 51 -0.515017851 -1.12564452 0.502055258  
## 52 0.896473726 0.74415578 0.945312760  
## 53 0.642841754 -0.16628565 -0.170279380  
## 54 -0.178693747 1.15472935 -0.043561781  
## 55 -1.424767642 1.89452559 0.097563001  
## 56 1.112213727 -0.35742907 1.296225745  
## 57 1.835606872 -0.08053937 -0.649425199  
## 58 -1.177246272 0.76869063 -1.232717630  
## 59 -0.005648087 -0.39408951 -0.403190239  
## 60 1.150470340 0.34495593 0.265491424  
## 61 -0.902863045 0.44275965 0.304197974  
## 62 -1.119184504 -0.99833108 0.382847311  
## 63 0.429549830 -0.55306830 -1.852042896  
## 64 -0.164027821 1.69662627 -2.752725605  
## 65 2.538162256 -0.65418206 0.791853284  
## 66 -0.880471699 0.24223639 -0.161662147  
## 67 1.013340822 0.96102595 0.981765651  
## 68 -1.008295632 1.36126985 0.114204349  
## 69 -0.343124605 -0.48441174 0.046498581  
## 70 -1.063279674 0.18652717 0.589714645  
## 71 1.224389557 0.11712625 -0.338387947  
## 72 0.245046510 -1.03056422 -0.811497312  
## 73 -0.471511697 1.12214360 0.350905299  
## 74 -1.531275253 -1.69116424 0.055797003  
## 75 -0.273409301 -0.84645543 -2.482558654  
## 76 -1.364005502 1.57236791 -0.250440488  
## 77 0.074217014 1.05084371 -0.874071321  
## 78 1.350920932 -0.44087858 0.786177958  
## 79 0.203589983 0.49216095 -2.146910373  
## 80 0.574842713 -0.45980488 0.616467340  
## 81 2.799989257 -0.19937910 -1.747462382  
## 82 0.924556986 -0.20532408 0.502878115  
## 83 -1.335624293 0.28601638 -0.480763384  
## 84 0.053562250 1.60970471 -0.050277120  
## 85 -1.234695245 -2.16655904 0.135823060  
## 86 0.730810035 -0.15897282 -1.968897844  
## 87 -0.342110395 -0.30665665 -0.071967528  
## 88 0.196579685 1.65450663 -0.450482378  
## 89 1.563831702 0.10439100 -0.269962887  
## 90 -0.570324975 0.74072535 0.351623347  
## 91 -1.440661951 -1.68564399 -0.034934452  
## 92 1.181854010 0.76241429 0.451426252  
## 93 -1.185871711 -1.09867306 -0.049116739  
## 94 -0.184728511 -1.03729527 1.180548765  
## 95 0.487942606 0.14581411 0.232424960  
## 96 0.276448381 -0.33797965 0.271770944  
## 97 -0.224408959 1.15721949 -1.169462467  
## 98 -0.186779822 -0.39740318 -0.072580194  
## 99 -0.378836256 -0.40329102 -1.691964004  
## 100 -0.867150698 1.53019187 -0.961666199  
## 101 1.709859594 0.18636579 0.913474571  
## 102 0.912341161 -0.37339222 -1.487642500  
## 103 0.468942474 -0.53925969 0.095604397  
## 104 1.037890507 -0.47686620 0.719858710  
## 105 0.942258412 0.97733531 -0.261258211  
## 106 2.468413555 0.92685473 -0.355816371  
## 107 0.735547777 0.04471160 1.797834220  
## 108 2.583677034 0.01601177 0.715299076  
## 109 1.983715711 -0.38158866 0.118116597  
## 110 0.430384846 -0.18385280 0.363334549  
## 111 -0.716963852 0.42618512 0.619894445  
## 112 1.385821140 0.51089598 -0.491139268  
## 113 1.599295536 -0.64894658 0.242640535  
## 114 2.394702152 -1.10651609 0.984931468  
## 115 1.398602423 0.32672611 -2.217900119  
## 116 1.443144452 0.46577704 0.715017254  
## 117 0.155051531 0.68789752 1.939011885  
## 118 1.323530516 1.07872343 0.377980452  
## 119 -0.116002279 1.50908972 1.190120490  
## 120 0.872557129 -0.29194848 -1.073766357  
## 121 1.361109864 0.64080977 0.153873302  
## 122 0.889438550 0.47957629 0.730093724  
## 123 1.945402229 -0.16906742 1.570657552  
## 124 1.811107380 -0.49629000 1.996388901  
## 125 -0.062549195 -1.43691748 -0.300588337  
## 126 2.382899631 0.51989558 -0.402716453  
## 127 1.694087190 0.75896938 -0.976941437  
## 128 0.953771626 -0.47311283 -0.537666028  
## 129 1.944108778 0.15411790 -0.058186942  
## 130 2.208012438 0.26236160 1.315507994  
## 131 2.354461380 -0.25014752 1.315676880  
## 132 -0.602606964 -1.68160247 -0.657961355  
## 133 1.349228024 0.37911553 -0.189784822  
## 134 1.418741031 0.87588247 0.005135135  
## 135 1.199659533 0.55930169 0.163268069  
## 136 0.367814360 -0.58389421 1.626853637  
## 137 2.347814225 -0.74613509 1.139091054  
## 138 0.302305637 -2.31771934 -0.070933376  
## 139 -1.368175141 0.14704595 -1.826921098  
## 140 1.218907551 -0.95833604 -2.301174512  
## 141 0.603462741 0.49927085 -1.727851893  
## 142 1.201200395 -0.23672771 -1.668092790  
## 143 2.236395763 -1.34988847 0.954506458  
## 144 -0.220303900 0.39300517 1.747584819  
## 145 2.269276612 -0.01271739 -0.631756964  
## 146 1.745788981 0.37494899 -1.526197986  
## 147 0.788021403 -0.64505131 0.012660587  
## 148 0.937757882 0.85334910 -0.112009896  
## 149 2.249150698 -1.27161336 0.454272354  
## 150 0.150001413 -0.38426111 -0.944526069  
## 151 -0.157035375 -1.08849456 0.467290992  
## 152 0.066274837 -2.52458577 -0.655495520  
## 153 0.649381101 -1.42066970 1.185829411  
## 154 -0.407280627 -0.77688046 -0.599581655  
## 155 -1.216924832 -3.34154057 0.212432192  
## 156 0.777457157 -0.28512773 1.292992506  
## 157 -1.018467262 0.07385763 0.768192685  
## 158 -0.193015687 -2.02200187 0.263321277  
## 159 -0.034532015 -1.50479254 0.408228999  
## 160 -2.644104155 -2.25029411 -0.191150566  
## 161 -1.501007268 -1.88133943 -0.127518980  
## 162 -0.403227218 0.27417984 0.401491580  
## 163 -1.907195211 -2.26184946 -1.476769844  
## 164 0.126091047 -0.87534790 -0.107636222  
## 165 0.735154480 -0.23350773 -0.719350737  
## 166 -1.037255484 0.13140673 0.421126511  
## 167 0.539376935 0.66580793 1.057593038  
## 168 -0.914459606 -1.50256385 0.198429555  
## 169 0.593660679 0.32242370 1.004488426  
## 170 0.121666760 -2.88390266 1.127059500  
## 171 -3.011645229 -0.36820566 1.171537403  
## 172 -1.428896595 0.03114149 1.359121024  
## 173 -0.231488767 -1.74730977 0.422941704  
## 174 -0.890130532 -2.81369612 -0.187297267  
## 175 -0.060286675 0.69757500 -0.134767564  
## 176 -1.258802488 -2.30505331 -0.875233596  
## 177 -0.744390359 -0.24506193 1.815765124  
## 178 -2.739240829 0.69329151 1.869358375  
## 179 -2.768394031 -0.17070832 -1.007206211  
## 180 1.051732696 -0.88266139 0.250290509  
## 181 -1.184224816 0.89236023 1.584707180  
## 182 0.114857474 -2.50565763 -0.751938064  
## 183 -1.118552867 -1.63186469 -1.626189689  
## 184 0.930903565 -2.27044144 0.304112108  
## 185 -2.434653564 -0.42622989 1.276621821  
## 186 0.180079441 0.17547193 -0.042755194  
## 187 -0.279323826 0.08447723 -1.686254589  
## 188 0.894506893 -1.19755634 0.788596636  
## 189 -0.615020833 1.20348186 0.554151722  
## 190 -2.564969241 -1.00246285 -2.880873055  
## 191 -0.507649696 -0.81564630 0.983182988  
## 192 -0.339798397 -0.74370285 0.105674804  
## 193 -1.126130720 0.10150397 -1.232893243  
## 194 -0.571323079 1.10337675 -0.990170484  
## 195 -0.995058839 -2.47377164 -0.486363052  
## 196 0.178417768 -0.82264537 0.266965159  
## 197 0.206372186 -0.16447686 -1.749486860  
## 198 -1.712961996 -2.10494756 0.468061380  
## 199 -0.409978783 0.53839117 -0.913500296  
## 200 0.111690810 -3.37409621 -0.562641734

Assign row names to the scores in landsat.lda:

rownames(landsat.pred$x) <- landsat$ct

Make a biplot of LD1 and LD2 scores, with the coefficients for the bands plotted as vectors:

biplot(landsat.pred$x, landsat.lda$scaling, cex = 0.5)

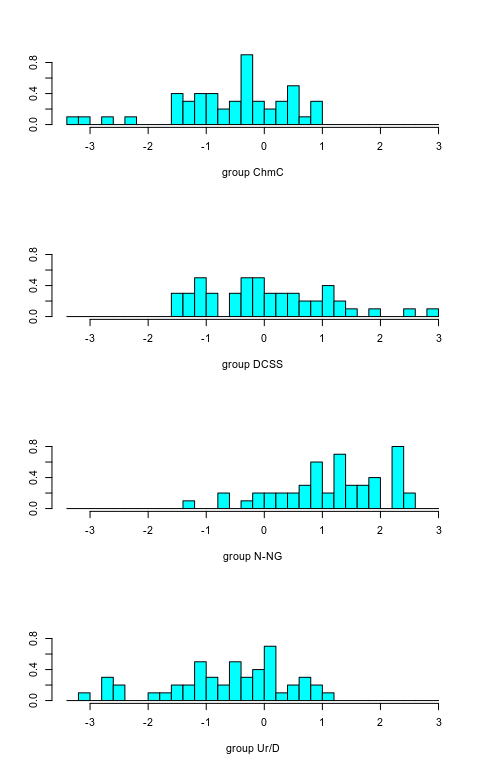


**Question: based on the coefficients, which variables are primarily responsible for the position of LD1? Which are responsible for the position of LD2?**

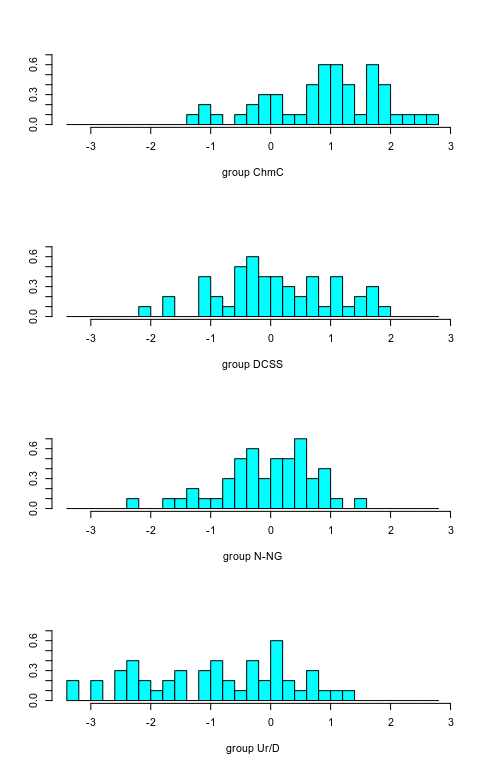
The variables that reach the furthest right or left are log.band7 and log.band5. The ones that extend furthest top to bottom are log.band1 and log.band2.

Make histograms for LD1 and LD2:

ldahist(landsat.pred$x[,1], landsat$ct)



ldahist(landsat.pred$x[,2], landsat$ct)



**Question: based on the histograms does it look like the cover types will separate? Do some separate better on LD1 than LD2?**

No, there is a lot of overlap in the histograms for both LD1 and LD2.

Calculate loadings:

cor(landsat[landsat.variables], landsat.pred$x) -> landsat.loadings  
  
landsat.loadings

## LD1 LD2 LD3  
## log.band1 -0.27868833 -0.7831109 -0.152428955  
## log.band2 -0.22690903 -0.7436151 -0.294126917  
## log.band3 -0.09025575 -0.6343877 -0.105575881  
## band4 -0.04650736 -0.5760724 0.011750666  
## log.band5 0.20397445 -0.3429468 0.007066303  
## band6 0.31722042 -0.6330620 0.002829578  
## log.band7 -0.13546143 -0.3201659 -0.027103626

**Question: which bands have the strongest correlation with LD1? Which have the strongest correlation with LD2?**

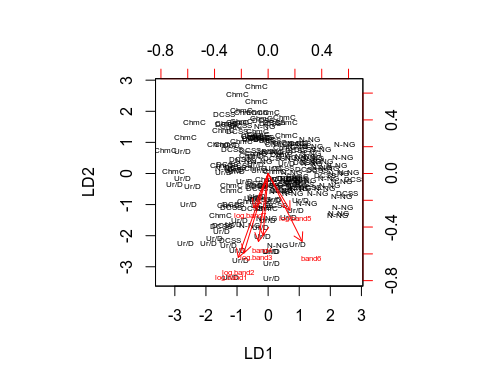
The strongest correlations with LD1 are band6 and log.band1. The strongest correlations with LD2 are log.band1, log.band2, log.band3, and band6.

**Question: all of the bands are negatively correlated with LD2. In what sense does this mean that LD2 is a “brightness” indicator?**

The reflectance is high for all of them when LD2 scores are low, so brighter pixels have low LD2 scores and dark pixels have high LD2 scores.

Make a biplot of scores with loadings as the vectors:

biplot(landsat.pred$x, landsat.loadings, cex = 0.5)



**Question: the biplot using loadings is quite different from the biplot using coefficients. Why?**

The coefficients are like regression slopes, and they are estimated as a group. Correlations between the variables are accounted for, and this can change the sign of the coefficients. Loadings are calculated one at a time, and the correlations between the bands are not accounted for. The loadings are better for understanding what pixels with high or low scores on LD1 or LD2 would look like, but the coefficients are used to calculate the scores in the first place.

Make a confustion matrix:

table(landsat.pred$class, landsat$ct, dnn=c("Predicted","Observed")) -> landsat.confusion  
landsat.confusion

## Observed  
## Predicted ChmC DCSS N-NG Ur/D  
## ChmC 31 15 4 9  
## DCSS 8 12 6 6  
## N-NG 5 14 37 5  
## Ur/D 6 9 3 30

**Question: do the cover types separate completely, or do they have overlapping values on tha seven bands? How do you know?**

No, they are overlapping because there are a lot of mis-classifications.

Calculate Cohen’s kappa using your own function.

First, source your cohens\_kappa.R file:

source("cohens\_kappa.R")

Calcualte kappa using your function:

kappa.biol532(landsat.confusion)

## [1] 0.4

**Question: is this considered a good value for kappa? To get a higher value what would we need to be true about the amount of separation between the cover types?**

No, this means we’re only doing 40% better than random chance. We would like kappa to be 0.7 or higher.

## Abledu paper

Import the foot data:

data.frame(read\_excel("footprint\_dimensions.xlsx")) -> feet

Split the data into left and right data sets:

left <- feet[ , c("sex","T1.l","T2.l","T3.l","T4.l","T5.l","BAB.l","BAH.l","Index.l")]  
right <- feet[ , c("sex","T1.r","T2.r","T3.r","T4.r","T5.r","BAB.r","BAH.r","Index.r")]

Extract the complete cases and replace the incomplete data with the complete data:

left <- left[complete.cases(left), ]  
  
right <- right[complete.cases(right), ]

Make a list for the lda models:

feet.lda <- list()

Conduct the same LDA’s that Abledu et al. used:

feet.lda$abledu.left.lda <- lda(sex ~ T5.l + BAH.l, data = left)  
feet.lda$abledu.right.lda <- lda(sex ~ T1.r + BAB.r + BAH.r, data = right)

Use all of the variables (all right variables for one LDA, all left variables for the other):

feet.lda$all.left.lda <- lda(sex ~ T1.l + T2.l + T3.l + T4.l + T5.l + BAH.l + BAB.l + Index.l, data = left)  
feet.lda$all.right.lda <- lda(sex ~ T1.r + T2.r + T3.r + T4.r + T5.r + BAH.r + BAB.r + Index.r, data = right)

Predict all four of your LDA models:

lapply(feet.lda, FUN = function(x) predict(x)) -> feet.predictions

Add the observed sexes for each model:

feet.predictions$abledu.left.lda$class.observed <- left$sex  
feet.predictions$abledu.right.lda$class.observed <- right$sex  
feet.predictions$all.left.lda$class.observed <- left$sex  
feet.predictions$all.right.lda$class.observed <- right$sex

Get a confusion matrix for each model:

lapply(feet.predictions, FUN = function(x) table(x$class, x$class.observed, dnn = c("Predicted","Observed"))) -> feet.confusion  
  
feet.confusion

## $abledu.left.lda  
## Observed  
## Predicted female male  
## female 44 11  
## male 12 55  
##   
## $abledu.right.lda  
## Observed  
## Predicted female male  
## female 45 13  
## male 14 53  
##   
## $all.left.lda  
## Observed  
## Predicted female male  
## female 45 9  
## male 11 57  
##   
## $all.right.lda  
## Observed  
## Predicted female male  
## female 43 12  
## male 16 54

Classification success for each model:

lapply(feet.confusion, FUN = function(x) sum(diag(x))/sum(x))

## $abledu.left.lda  
## [1] 0.8114754  
##   
## $abledu.right.lda  
## [1] 0.784  
##   
## $all.left.lda  
## [1] 0.8360656  
##   
## $all.right.lda  
## [1] 0.776