Linear Discriminant Analysis

Loading required libraries

```
library(cluster)
library(data.table)
library(magrittr)
library(stringr)
library(ggplot2)
library(knitr)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------- tidyverse 1.3.0 --

## v tibble  3.0.3     v purrr   0.3.4
## v tidyr   1.1.2     v dplyr   1.0.2
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts -------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::between()   masks data.table::between()
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x purrr::set_names() masks magrittr::set_names()
## x purrr::transpose() masks data.table::transpose()
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
library(FactoMineR)
library(nFactors)
```

```
## Loading required package: lattice

##
## Attaching package: 'nFactors'

## The following object is masked from 'package:lattice':
##
##     parallel
```

```r
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'

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

```r
library(gvlma)
library(leaps)
library(relaimpo)
```

```
## Loading required package: boot

##
## Attaching package: 'boot'

## The following object is masked from 'package:lattice':
##
##     melanoma

## The following object is masked from 'package:psych':
##
##     logit

## Loading required package: survey

## Loading required package: grid
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:boot':
##
##     aml

##
## Attaching package: 'survey'

## The following object is masked from 'package:graphics':
##
##     dotchart

## Loading required package: mitools

## This is the global version of package relaimpo.

## If you are a non-US user, a version with the interesting additional metric pmvd is available

## from Ulrike Groempings web site at prof.beuth-hochschule.de/groemping.
```

```r
library(cowplot)
library(regclass)
```

```
## Loading required package: bestglm

## Loading required package: VGAM

## Loading required package: stats4

## Loading required package: splines

##
## Attaching package: 'VGAM'

## The following object is masked from 'package:survey':
##
##     calibrate
```

```
## The following objects are masked from 'package:boot':
##
##     logit, simplex

## The following objects are masked from 'package:psych':
##
##     fisherz, logistic, logit

## The following object is masked from 'package:tidyr':
##
##     fill

## Loading required package: rpart

## Loading required package: randomForest

## randomForest 4.6-14

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

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:psych':
##
##     outlier

## The following object is masked from 'package:dplyr':
##
##     combine

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

## Important regclass change from 1.3:
## All functions that had a . in the name now have an _
## all.correlations -> all_correlations, cor.demo -> cor_demo, etc.

##
## Attaching package: 'regclass'

## The following object is masked from 'package:lattice':
##
##     qq
```

```r
library(e1071)
library(caret)
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:VGAM':
##
##     predictors

## The following object is masked from 'package:survival':
##
##     cluster

## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(ROCR)
library(klaR)
#library(FFally)
```

Data Loading

```
Lending_Data <- read_csv('Lending_Data.csv')
```

```
## Parsed with column specification:
## cols(
##   member_id = col_character(),
##   loan_status = col_character(),
##   int_rate = col_character(),
##   Bin_int = col_double(),
##   dti = col_double(),
##   Bin_dti = col_double(),
##   Default_flag = col_double(),
##   No_of_Enquiry = col_double(),
##   enq_buckets = col_character(),
##   annual_inc = col_double(),
##   Income_bins = col_double(),
##   home_ownership = col_character(),
##   purpose = col_character(),
##   open_acc = col_double(),
##   emp_length = col_character(),
##   verification_status = col_character(),
```

```
##   delinq_2yrs = col_double(),
##   loan_amnt = col_double(),
##   Bins_loan_amt = col_double()
## )
```

```
Lend = copy(Lending_Data)
Lend = setDT(Lend)
#view(Lend)
str(Lend)
```

```
## Classes 'data.table' and 'data.frame':   35808 obs. of  19 variables:
##  $ member_id          : chr  "LC1" "LC10" "LC100" "LC1000" ...
##  $ loan_status        : chr  "Charged Off" "Fully Paid" "Fully Paid" "Fully Paid" ...
##  $ int_rate           : chr  "11.71%" "15.96%" "10.65%" "12.69%" ...
##  $ Bin_int            : num  10 16 8 11 22 1 23 10 5 16 ...
##  $ dti                : num  1.06 2.61 11.34 14 13.01 ...
##  $ Bin_dti            : num  2 3 11 14 13 11 5 10 24 14 ...
##  $ Default_flag       : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ No_of_Enquiry      : num  0 1 1 1 0 0 3 0 1 2 ...
##  $ enq_buckets        : chr  "0" "1-4" "1-4" "1-4" ...
##  $ annual_inc         : num  110000 135000 75000 51000 41500 ...
##  $ Income_bins        : num  9 11 6 4 3 4 12 7 6 4 ...
##  $ home_ownership     : chr  "MORTGAGE" "RENT" "MORTGAGE" "RENT" ...
##  $ purpose            : chr  "credit_card" "other" "educational" "credit_card" ...
##  $ open_acc           : num  6 3 7 5 8 5 4 7 6 9 ...
##  $ emp_length         : chr  "LT 1year" "10+ years" "2 years" "1 year" ...
##  $ verification_status: chr  "Not Verified" "Source Verified" "Source Verified" "Source Verified" ..
##  $ delinq_2yrs        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ loan_amnt          : num  7000 2000 12000 9350 6000 ...
##  $ Bins_loan_amt      : num  6 2 10 8 5 8 5 10 2 8 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   member_id = col_character(),
##   ..   loan_status = col_character(),
##   ..   int_rate = col_character(),
##   ..   Bin_int = col_double(),
##   ..   dti = col_double(),
##   ..   Bin_dti = col_double(),
##   ..   Default_flag = col_double(),
##   ..   No_of_Enquiry = col_double(),
##   ..   enq_buckets = col_character(),
##   ..   annual_inc = col_double(),
##   ..   Income_bins = col_double(),
##   ..   home_ownership = col_character(),
##   ..   purpose = col_character(),
##   ..   open_acc = col_double(),
##   ..   emp_length = col_character(),
##   ..   verification_status = col_character(),
##   ..   delinq_2yrs = col_double(),
##   ..   loan_amnt = col_double(),
##   ..   Bins_loan_amt = col_double()
##   .. )
##  - attr(*, ".internal.selfref")=<externalptr>
```

Data Cleaning

```
Lend[, member_id := factor(member_id)]
Lend[, loan_status := factor(loan_status)]
Lend[, home_ownership := factor(home_ownership)]
Lend[, purpose := factor(purpose)]
Lend[, verification_status := factor(verification_status)]

Lend[, int_rate := gsub('[%]', '', int_rate)]
Lend[, int_rate := trimws(int_rate)]
Lend[, int_rate := suppressWarnings(as.numeric(int_rate))]

Lend[open_acc %in% c(1, 2, 3, 4, 5), 'x' := 'LT5']
Lend[open_acc %in% c(6, 7, 8, 9, 10), 'x' := '6-10']
Lend[open_acc %in% c(11, 12, 13, 14, 15), 'x' := '11-15']
Lend[open_acc > 15, 'x' := '15+']
Lend = Lend %>% rename(no_of_acct = x)
str(Lend)
```

```
## Classes 'data.table' and 'data.frame':   35808 obs. of  20 variables:
##  $ member_id          : Factor w/ 35808 levels "LC1","LC10","LC100",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ loan_status        : Factor w/ 2 levels "Charged Off",..: 1 2 2 2 2 2 2 2 2 2 ...
##  $ int_rate           : num  11.7 16 10.7 12.7 19.7 ...
##  $ Bin_int            : num  10 16 8 11 22 1 23 10 5 16 ...
##  $ dti                : num  1.06 2.61 11.34 14 13.01 ...
##  $ Bin_dti            : num  2 3 11 14 13 11 5 10 24 14 ...
##  $ Default_flag       : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ No_of_Enquiry      : num  0 1 1 1 0 0 3 0 1 2 ...
##  $ enq_buckets        : chr  "0" "1-4" "1-4" "1-4" ...
##  $ annual_inc         : num  110000 135000 75000 51000 41500 ...
##  $ Income_bins        : num  9 11 6 4 3 4 12 7 6 4 ...
##  $ home_ownership     : Factor w/ 5 levels "MORTGAGE","NONE",..: 1 5 1 5 1 1 1 5 5 1 ...
##  $ purpose            : Factor w/ 14 levels "car","credit_card",..: 2 10 4 2 3 3 8 2 10 3 ...
##  $ open_acc           : num  6 3 7 5 8 5 4 7 6 9 ...
##  $ emp_length         : chr  "LT 1year" "10+ years" "2 years" "1 year" ...
##  $ verification_status: Factor w/ 3 levels "Not Verified",..: 1 2 2 2 3 3 1 1 1 2 ...
##  $ delinq_2yrs        : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ loan_amnt          : num  7000 2000 12000 9350 6000 ...
##  $ Bins_loan_amt      : num  6 2 10 8 5 8 5 10 2 8 ...
##  $ no_of_acct         : chr  "6-10" "LT5" "6-10" "LT5" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   member_id = col_character(),
##   ..   loan_status = col_character(),
##   ..   int_rate = col_character(),
##   ..   Bin_int = col_double(),
##   ..   dti = col_double(),
##   ..   Bin_dti = col_double(),
##   ..   Default_flag = col_double(),
##   ..   No_of_Enquiry = col_double(),
##   ..   enq_buckets = col_character(),
##   ..   annual_inc = col_double(),
##   ..   Income_bins = col_double(),
```

```
##   ..   home_ownership = col_character(),
##   ..   purpose = col_character(),
##   ..   open_acc = col_double(),
##   ..   emp_length = col_character(),
##   ..   verification_status = col_character(),
##   ..   delinq_2yrs = col_double(),
##   ..   loan_amnt = col_double(),
##   ..   Bins_loan_amt = col_double()
##   .. )
##  - attr(*, ".internal.selfref")=<externalptr>
##  - attr(*, "index")= int
##   ..- attr(*, "__open_acc")= int  75 113 157 195 377 382 458 611 628 642 ...
```

```
Lend_lda <- read_csv('Logistic_training_final.csv')
```

```
## Parsed with column specification:
## cols(
##   loan_status = col_double(),
##   roi = col_double(),
##   loan_amnt = col_double(),
##   inq_last_6mths = col_character(),
##   purpose = col_character(),
##   revol_util = col_double(),
##   Late_fee_bin = col_character(),
##   term = col_double(),
##   total_pymnt = col_double()
## )
```

```
Lend_lda = setDT(Lend_lda)
```

Data Splitting

```
#Training Testing

## 10% of the sample size
smp_size = floor(0.50 * nrow(Lend_lda))

## set the seed to make our partition reproducible
set.seed(123)
train_ind = sample(seq_len(nrow(Lend_lda)), size = smp_size)
head(train_ind)
```

```
## [1]  2986 29925 29710 37529  2757 38938
```

```
train = Lend_lda[train_ind, ]
test = Lend_lda[-train_ind, ]
head(train)
```

```
##    loan_status  roi loan_amnt inq_last_6mths            purpose revol_util
## 1:           0 0.13     10000           zero debt_consolidation       0.87
```

```
## 2:           0 0.15      5400           zero debt_consolidation       0.95
## 3:           1 0.12      3000          three              other       0.66
## 4:           0 0.09      6000            one   home_improvement       0.63
## 5:           0 0.08      9600           zero debt_consolidation       0.20
## 6:           0 0.12     13000           zero debt_consolidation       0.46
##    Late_fee_bin term total_pymnt
## 1:            0   36   11568.167
## 2:          GT1   36    6783.276
## 3:            0   36     994.100
## 4:            0   36    6858.690
## 5:            0   36   10751.870
## 6:            0   60   16414.700
```

```r
head(test)
```

```
##    loan_status  roi loan_amnt inq_last_6mths            purpose revol_util
## 1:           1 0.08      3800          three                car       0.39
## 2:           0 0.06      7500           zero            medical       0.36
## 3:           0 0.22     24625            one debt_consolidation       0.95
## 4:           0 0.06      5000           zero debt_consolidation       0.14
## 5:           0 0.20     12000           zero debt_consolidation       0.89
## 6:           0 0.08      6000           zero              other       0.23
##    Late_fee_bin term total_pymnt
## 1:            0   36    1064.070
## 2:            0   36    7835.776
## 3:            0   60   31696.993
## 4:            0   36    5478.388
## 5:            0   60   13766.134
## 6:            0   36    6534.334
```

Linear Discriminant Analysis

```r
str(train)
```

```
## Classes 'data.table' and 'data.frame':   19893 obs. of  9 variables:
##  $ loan_status   : num  0 0 1 0 0 0 0 0 0 0 ...
##  $ roi           : num  0.13 0.15 0.12 0.09 0.08 0.12 0.14 0.12 0.07 0.13 ...
##  $ loan_amnt     : num  10000 5400 3000 6000 9600 13000 12000 13000 9500 2100 ...
##  $ inq_last_6mths: chr  "zero" "zero" "three" "one" ...
##  $ purpose       : chr  "debt_consolidation" "debt_consolidation" "other" "home_improvement" ...
##  $ revol_util    : num  0.87 0.95 0.66 0.63 0.2 0.46 0.54 0.09 0.56 0.93 ...
##  $ Late_fee_bin  : chr  "0" "GT1" "0" "0" ...
##  $ term          : num  36 36 36 36 36 60 36 36 36 36 ...
##  $ total_pymnt   : num  11568 6783 994 6859 10752 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   loan_status = col_double(),
##   ..   roi = col_double(),
##   ..   loan_amnt = col_double(),
##   ..   inq_last_6mths = col_character(),
```

```
##   ..   purpose = col_character(),
##   ..   revol_util = col_double(),
##   ..   Late_fee_bin = col_character(),
##   ..   term = col_double(),
##   ..   total_pymnt = col_double()
##   .. )
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
lend_lda <- lda(formula = loan_status ~ ., data = Lend_lda)
```

```r
summary(lend_lda)
```

```
##         Length Class  Mode
## prior    2     -none- numeric
## counts   2     -none- numeric
## means   56     -none- numeric
## scaling 28     -none- numeric
## lev      2     -none- character
## svd      1     -none- numeric
## N        1     -none- numeric
## call     3     -none- call
## terms    3     terms  call
## xlevels  3     -none- list
```

```r
print(lend_lda)
```

```
## Call:
## lda(loan_status ~ ., data = Lend_lda)
##
## Prior probabilities of groups:
##         0         1
## 0.8574876 0.1425124
##
## Group means:
##         roi loan_amnt inq_last_6mthsfive inq_last_6mthsfour inq_last_6mthsone
## 0 0.1171957  11079.10        0.003253605        0.008265916         0.2923848
## 1 0.1381376  12147.49        0.004409171        0.009523810         0.3287478
##   inq_last_6mthsseven inq_last_6mthssix inq_last_6mthsthree inq_last_6mthstwo
## 0         0.000732794       0.001406964          0.06521867         0.1290597
## 1         0.001940035       0.002821869          0.09894180         0.1472663
##   inq_last_6mthszero purposecredit_card purposedebt_consolidation
## 0          0.4993258         0.13451167                 0.4655880
## 1          0.4059965         0.09664903                 0.4924162
##   purposeeducational purposehome_improvement purposehouse purposemajor_purchase
## 0        0.007884863              0.07720718  0.009467698            0.05762692
## 1        0.009876543              0.06190476  0.010405644            0.03915344
##   purposemedical purposemoving purposeother purposerenewable_energy
## 0     0.01726463    0.01439207   0.09860476             0.002462188
## 1     0.01869489    0.01622575   0.11234568             0.003350970
##   purposesmall_business purposevacation purposewedding revol_util
## 0            0.03962950     0.009614257     0.02497362  0.4773660
## 1            0.08447972     0.009347443     0.01693122  0.5552081
```

```
##   Late_fee_bin0to1 Late_fee_binGT1     term total_pymnt
## 0     0.0002638058      0.03505686 41.80162   13103.416
## 1     0.0003527337      0.15361552 46.34074    6988.959
##
## Coefficients of linear discriminants:
##                                   LD1
## roi                      9.7543953914
## loan_amnt                0.0002997093
## inq_last_6mthsfive       0.7929512451
## inq_last_6mthsfour       0.9116960505
## inq_last_6mthsone        0.8591832093
## inq_last_6mthsseven      1.2230444382
## inq_last_6mthssix        1.2851973870
## inq_last_6mthsthree      0.9330229414
## inq_last_6mthstwo        0.8344431328
## inq_last_6mthszero       0.8201722976
## purposecredit_card       0.0400383429
## purposedebt_consolidation  0.1262749863
## purposeeducational       0.2315900323
## purposehome_improvement  0.0658657976
## purposehouse             0.1429374485
## purposemajor_purchase    0.0033596459
## purposemedical           0.1330898738
## purposemoving            0.1729571637
## purposeother             0.1611993949
## purposerenewable_energy  0.2135061571
## purposesmall_business    0.2783123520
## purposevacation          0.1647687318
## purposewedding          -0.0241097097
## revol_util               0.2284524638
## Late_fee_bin0to1         0.4313335472
## Late_fee_binGT1          1.1810440788
## term                     0.0203617252
## total_pymnt             -0.0002873950
```

```
lend_lda$counts
```

```
##     0     1
## 34116  5670
```

```
lend_lda$means
```

```
##         roi loan_amnt inq_last_6mthsfive inq_last_6mthsfour inq_last_6mthsone
## 0 0.1171957  11079.10        0.003253605        0.008265916         0.2923848
## 1 0.1381376  12147.49        0.004409171        0.009523810         0.3287478
##   inq_last_6mthsseven inq_last_6mthssix inq_last_6mthsthree inq_last_6mthstwo
## 0         0.000732794       0.001406964          0.06521867         0.1290597
## 1         0.001940035       0.002821869          0.09894180         0.1472663
##   inq_last_6mthszero purposecredit_card purposedebt_consolidation
## 0          0.4993258         0.13451167                 0.4655880
## 1          0.4059965         0.09664903                 0.4924162
##   purposeeducational purposehome_improvement purposehouse purposemajor_purchase
## 0        0.007884863              0.07720718  0.009467698            0.05762692
```

```
## 1        0.009876543              0.06190476  0.010405644            0.03915344
##   purposemedical purposemoving purposeother purposerenewable_energy
## 0     0.01726463    0.01439207   0.09860476             0.002462188
## 1     0.01869489    0.01622575   0.11234568             0.003350970
##   purposesmall_business purposevacation purposewedding revol_util
## 0            0.03962950     0.009614257     0.02497362  0.4773660
## 1            0.08447972     0.009347443     0.01693122  0.5552081
##   Late_fee_bin0to1 Late_fee_binGT1     term total_pymnt
## 0     0.0002638058      0.03505686 41.80162   13103.416
## 1     0.0003527337      0.15361552 46.34074    6988.959
```

```
lend_lda$scaling
```

```
##                                     LD1
## roi                        9.7543953914
## loan_amnt                  0.0002997093
## inq_last_6mthsfive         0.7929512451
## inq_last_6mthsfour         0.9116960505
## inq_last_6mthsone          0.8591832093
## inq_last_6mthsseven        1.2230444382
## inq_last_6mthssix          1.2851973870
## inq_last_6mthsthree        0.9330229414
## inq_last_6mthstwo          0.8344431328
## inq_last_6mthszero         0.8201722976
## purposecredit_card         0.0400383429
## purposedebt_consolidation  0.1262749863
## purposeeducational         0.2315900323
## purposehome_improvement    0.0658657976
## purposehouse               0.1429374485
## purposemajor_purchase      0.0033596459
## purposemedical             0.1330898738
## purposemoving              0.1729571637
## purposeother               0.1611993949
## purposerenewable_energy    0.2135061571
## purposesmall_business      0.2783123520
## purposevacation            0.1647687318
## purposewedding            -0.0241097097
## revol_util                 0.2284524638
## Late_fee_bin0to1           0.4313335472
## Late_fee_binGT1            1.1810440788
## term                       0.0203617252
## total_pymnt               -0.0002873950
```

```
lend_lda$prior
```

```
##         0         1
## 0.8574876 0.1425124
```

```
lend_lda$lev
```

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

```r
lend_lda$svd
```

```
## [1] 178.2008
```

Singular values (svd) that gives the ratio of the between- and within-group standard deviations on the linear discriminant variables.

```r
class(lend_lda)
```

```
## [1] "lda"
```

```r
#?lda
lend_lda$N
```

```
## [1] 39786
```

```r
lend_lda$call
```

```
## lda(formula = loan_status ~ ., data = Lend_lda)
```

```r
(prop = lend_lda$svd^2/sum(lend_lda$svd^2))
```

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

We can use the singular values to compute the amount of the between-group variance that is explained by each linear discriminant. In our example we see that the first linear discriminant explains more than 99% of the between-group variance in the lending dataset.

```r
lend_lda_2 <- lda(formula = loan_status ~ ., data = Lend_lda, CV = TRUE)
#lend_lda_2
head(lend_lda_2$class)
```

```
## [1] 1 0 0 1 1 0
## Levels: 0 1
```

The Maximum a Posteriori Probability (MAP) classification (a factor) posterior: posterior probabilities for the classes.

```r
head(lend_lda_2$posterior, 3)
```

```
##              0          1
## 1 1.988830e-07 0.99999980
## 2 9.476054e-01 0.05239458
## 3 9.619433e-01 0.03805669
```

```r
train <- sample(1:150, 75)
lend_lda_3 <- lda(loan_status ~ ., Lend_lda) # training model
lend_plda = predict(object = lend_lda_3, newdata = test) # predictions
```

```r
head(lend_plda$class)
```

```
## [1] 0 0 0 0 0 0
## Levels: 0 1
```

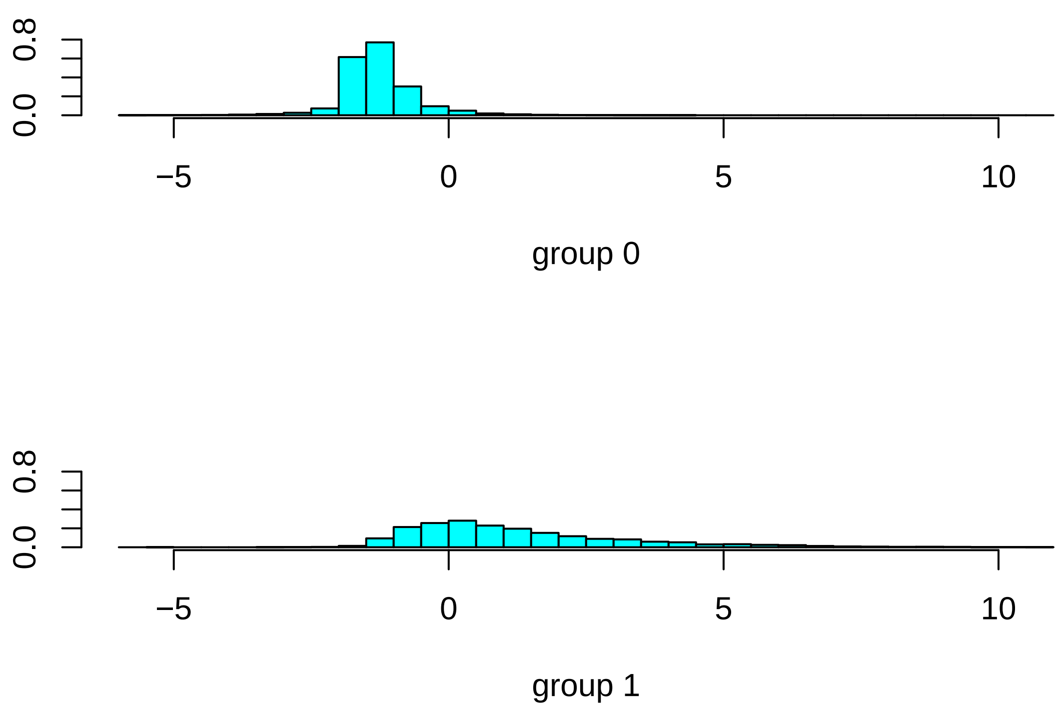
```r
head(lend_plda$posterior, 6) # posterior prob.
```

```
##           0           1
## 1 0.9614875 0.038512513
## 2 0.9970457 0.002954254
## 3 0.9897072 0.010292781
## 4 0.9978728 0.002127159
## 5 0.8454720 0.154527977
## 6 0.9960104 0.003989559
```

```r
head(lend_plda$x, 3)
```

```
##          LD1
## 1  0.3568485
## 2 -0.6620839
## 3 -0.1707899
```

```r
plot(lend_lda)
plot(lend_lda_3)
```



```
lend_lda <- lda(loan_status ~ ., Lend_lda)
prop_lda = lend_lda$svd^2/sum(lend_lda$svd^2)
lend_plda <- predict(object = lend_lda, newdata = Lend_lda)
dataset = data.frame(Defaulters = Lend_lda[, "loan_status"], lda = lend_plda$x)
head(dataset)
```

```
##   loan_status        LD1
## 1           1  7.6482734
## 2           0  0.4825096
## 3           1  0.3568485
## 4           1  4.6983544
## 5           1  1.9997528
## 6           0 -0.6620839
```

Lets play with accuracy lets look at another way to divide a dataset

```
set.seed(101) # Nothing is random!!
sample_n(Lend_lda, 10)
```

```
##     loan_status  roi loan_amnt inq_last_6mths            purpose revol_util
##  1:           0 0.12     11200           zero debt_consolidation       0.48
##  2:           0 0.07      5000           zero              other       0.10
##  3:           0 0.08     16700           zero        credit_card       0.28
##  4:           0 0.12      3000            two debt_consolidation       0.71
##  5:           0 0.08     12400           zero debt_consolidation       0.47
##  6:           0 0.06     10600           zero        credit_card       0.53
##  7:           0 0.08      4500           zero     small_business       0.10
##  8:           1 0.16     11200          three              other       0.94
##  9:           0 0.13      6000           zero debt_consolidation       0.69
## 10:           0 0.21     30000           zero debt_consolidation       0.85
##     Late_fee_bin term total_pymnt
##  1:            0   36   13469.140
##  2:            0   36    5532.340
##  3:          GT1   36   18728.829
##  4:            0   36    3579.662
##  5:            0   36   13133.028
##  6:            0   36   11414.441
##  7:            0   36    4962.064
##  8:            0   60    2647.710
##  9:            0   36    7302.734
## 10:            0   60   47346.040
```

Lets take a sample of 75/25 like before. Dplyr preserves class.

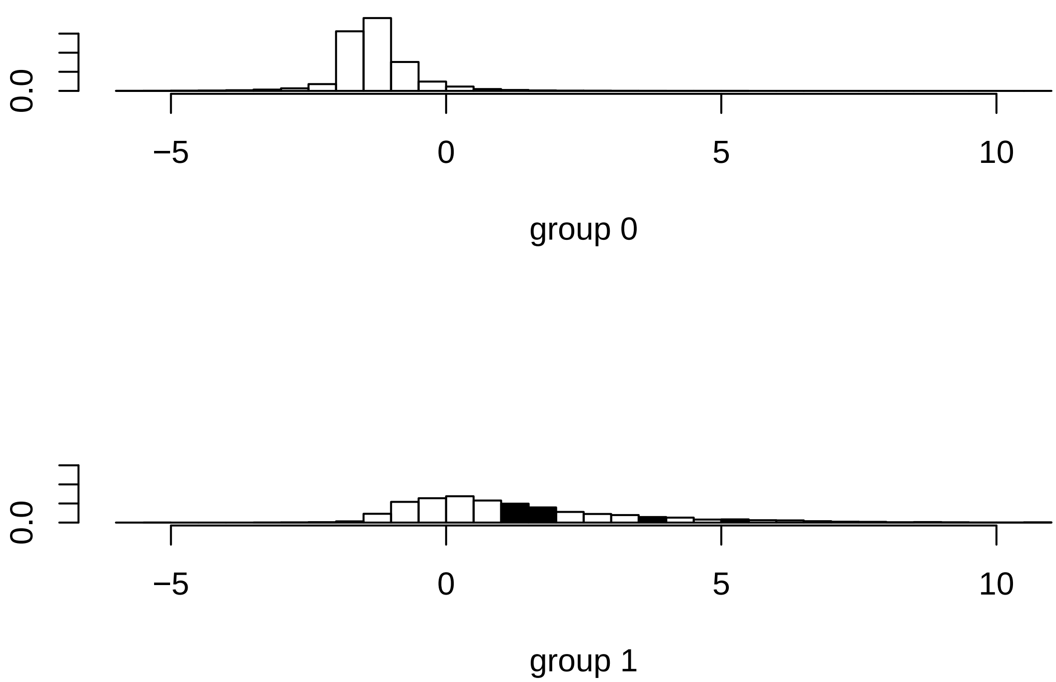
```
training_sample <- sample(c(TRUE, FALSE), nrow(Lend_lda), replace = T, prob = c(0.75,0.25))
train <- Lend_lda[training_sample, ]
test <- Lend_lda[!training_sample, ]
```

Lets run LDA like before

```r
lda_lend <- lda(loan_status ~ ., train)
```

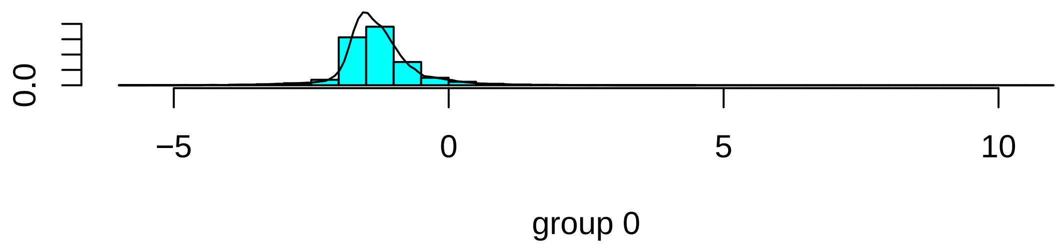
Do a quick plot to understand how good the model is

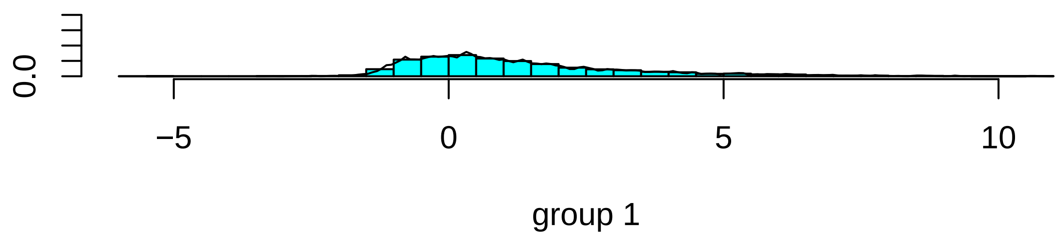
```r
plot(lda_lend, col = as.integer(train$loan_status))
```



Sometime bell curves are better

```r
plot(lda_lend, dimen = 1, type = "b")
```





This plot shows the essense of LDA. It puts everything on a line and finds cutoffs. Partition plots

```
#partimat(loan_status ~ ., data=train, method="lda")
```

Lets focus on accuracy. Table function

```
lda_train <- predict(lda_lend)
train$lda <- lda_train$class
table(train$lda, train$loan_status)
```

```
##
##         0     1
##   0 25127  2000
##   1   380  2167
```

Running accuracy on the training set shows how good the model is. It is not an indication of "true" accuracy. We will use the test set to approximate accuracy

```
lda_test <- predict(lda_lend, test)
test$lda <- lda_test$class
table(test$lda, test$loan_status)
```

```
##
##        0    1
##   0 8469  729
##   1  140  774
```