Lets cut the data into two parts
#75 % and 25%

smp_size_raw <- floor(0.75 * nrow(loan)) #Calculating a 75% sample from the population of 1370718 obs i.e 98038 observations
train_ind_raw <- sample(nrow(loan), size = smp_size_raw) #Taking a sample of size 98038 from the population of 1370718 observations
train_raw.df <- as.data.frame(loan[train_ind_raw,]) # first 98038 observations for the training
test_raw.df <- as.data.frame(loan[-train_ind_raw,]) #the remaining 32680 observations for testing the data

We now have a training and a test set. Training is 75% and test is 25%

```
> loan_raw.lda
Call:
lda(train_raw.df$loan_status ~ ., data = train_raw.df)

Prior probabilities of groups:
        0         1
0.2770485 0.7229515

Group means:
  loan_amnt funded_amnt installment int_rate  issue_d    grade   purpose      dti emp_length home_ownership
0  15955.75    15955.75    490.3159 15.93232 2016.265 4.598958 0.2311959 21.06700   6.289942      0.5771780
1  14059.15    14059.15    434.8547 12.90349 2016.209 5.339308 0.2316964 18.98579   6.405031      0.6639579
  annual_inc      term
0   78376.33 0.6590233
1   84852.53 0.8110476

Coefficients of linear discriminants:
                         LD1
loan_amnt       6.832713e-06
funded_amnt     6.832713e-06
installment    -1.064352e-03
int_rate       -8.285198e-02
issue_d        -8.809209e-02
grade           2.782836e-01
purpose        -1.076464e-01
dti            -1.307558e-02
emp_length      3.842267e-03
home_ownership  5.678624e-01
annual_inc      1.007924e-06
term            6.471866e-01
```

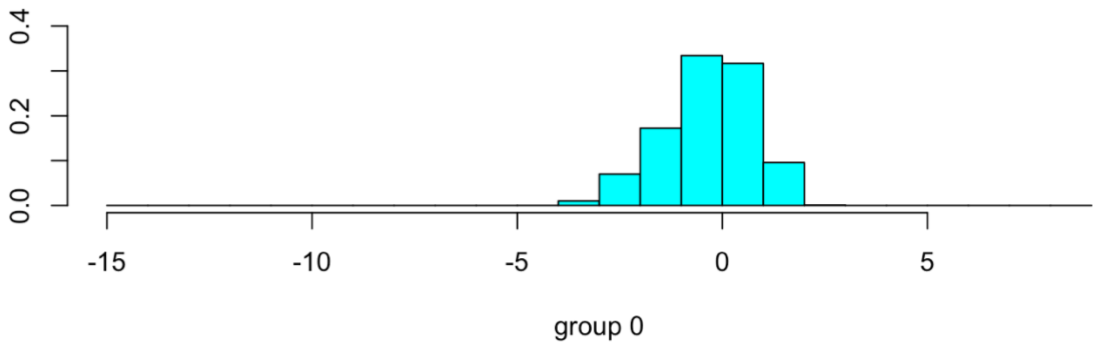
summary(loan_raw.lda)

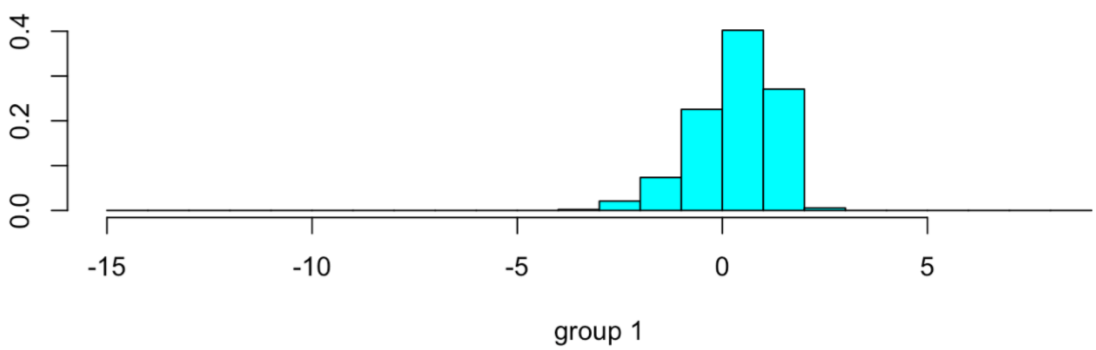
```
> summary(loan_raw.lda)
        Length Class  Mode
prior    2     -none- numeric
counts   2     -none- numeric
means   24     -none- numeric
scaling 12     -none- numeric
lev      2     -none- character
svd      1     -none- numeric
N        1     -none- numeric
call     3     -none- call
terms    3     terms  call
xlevels  0     -none- list
```

print(loan_raw.lda)

```
> print(loan_raw.lda)
Call:
lda(train_raw.df$loan_status ~ ., data = train_raw.df)

Prior probabilities of groups:
        0         1
0.2770485 0.7229515

Group means:
  loan_amnt funded_amnt installment int_rate  issue_d    grade   purpose      dti emp_length home_ownership
0  15955.75    15955.75    490.3159 15.93232 2016.265 4.598958 0.2311959 21.06700   6.289942      0.5771780
1  14059.15    14059.15    434.8547 12.90349 2016.209 5.339308 0.2316964 18.98579   6.405031      0.6639579
  annual_inc      term
0   78376.33 0.6590233
1   84852.53 0.8110476

Coefficients of linear discriminants:
                        LD1
loan_amnt       6.832713e-06
funded_amnt     6.832713e-06
installment    -1.064352e-03
int_rate       -8.285198e-02
issue_d        -8.809209e-02
grade           2.782836e-01
purpose        -1.076464e-01
dti            -1.307558e-02
emp_length      3.842267e-03
home_ownership  5.678624e-01
annual_inc      1.007924e-06
term            6.471866e-01
```

plot(loan_raw.lda)





loan_raw.lda.predict <- predict(loan_raw.lda, newdata = test_raw.df)
#prediction will be done on the test data which is 25% of the overall population
#the above prediction has three items called class, posterior and x

loan_raw.lda.predict$class #contains a list of 1's and 0's

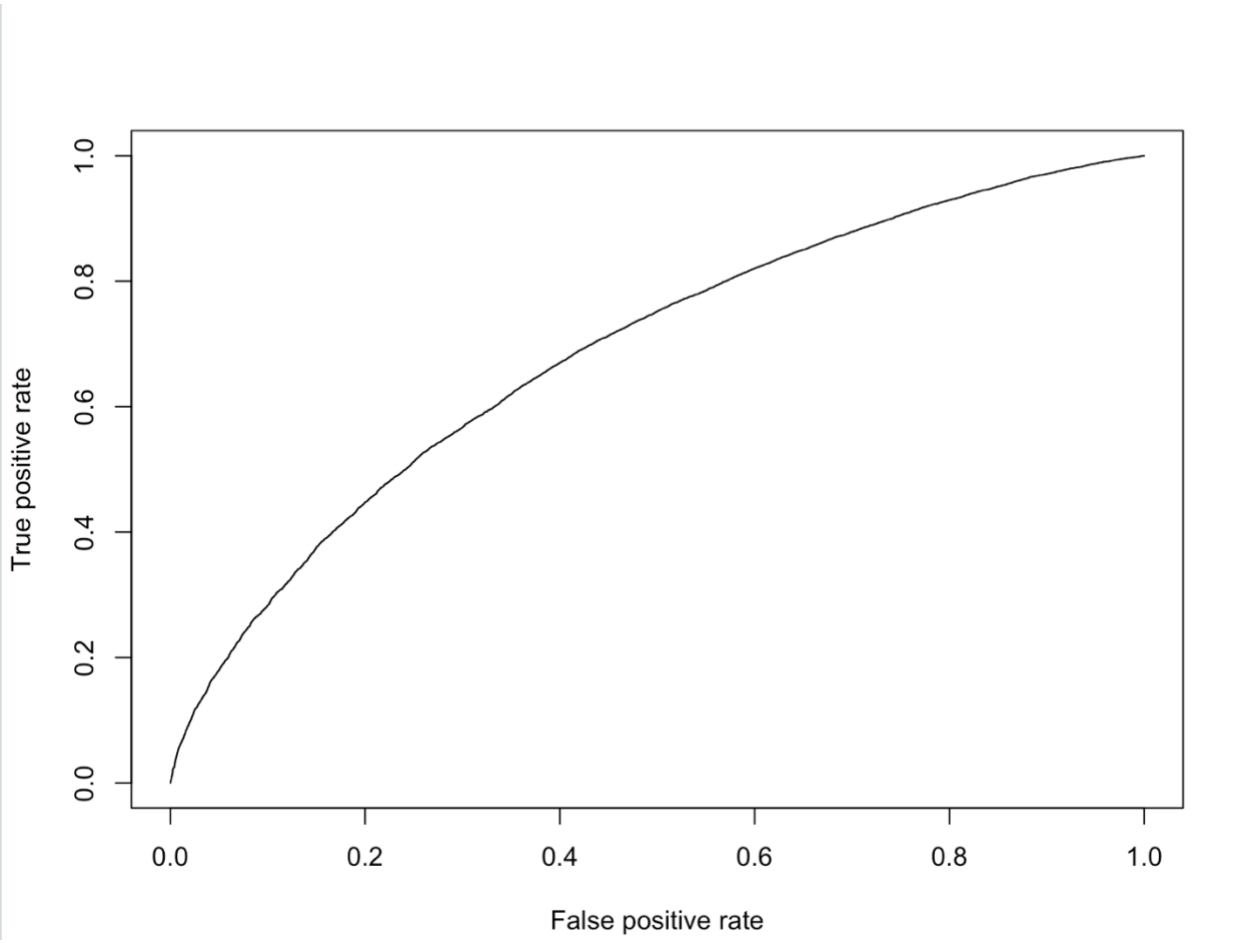
```
   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1
  [56] 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
 [111] 1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1
 [166] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [221] 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1
 [276] 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1
 [331] 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
 [386] 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1
 [441] 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 0 0 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1
 [496] 0 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1
 [551] 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [606] 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
 [661] 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [716] 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1 1 1 1 0 1
 [771] 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
 [826] 1 1 1 1 1 0 1 1 1 1 0 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 1 1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 1 1 1
 [881] 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0 1 0
 [936] 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [991] 1 1 0 1 1 1 0 1 1 1
 [ reached getOption("max.print") -- omitted 34576 entries ]
Levels: 0 1
```

loan_raw.lda.predict$x #all the z-scores

4	4.451842e-01
11	-1.539352e+00
19	2.236999e-01
20	-3.400426e-01
22	2.342337e-01
23	1.267913e+00
24	7.936621e-01
40	1.168131e+00
65	6.940014e-01
71	-2.831146e-02
92	8.470080e-01
93	-5.119140e-01
94	7.603107e-01
100	3.215108e-01
104	1.481849e+00
105	3.746894e-01
140	-3.185896e+00
149	7.757711e-01
172	1.402711e+00
190	6.012481e-01
226	-8.543132e-02

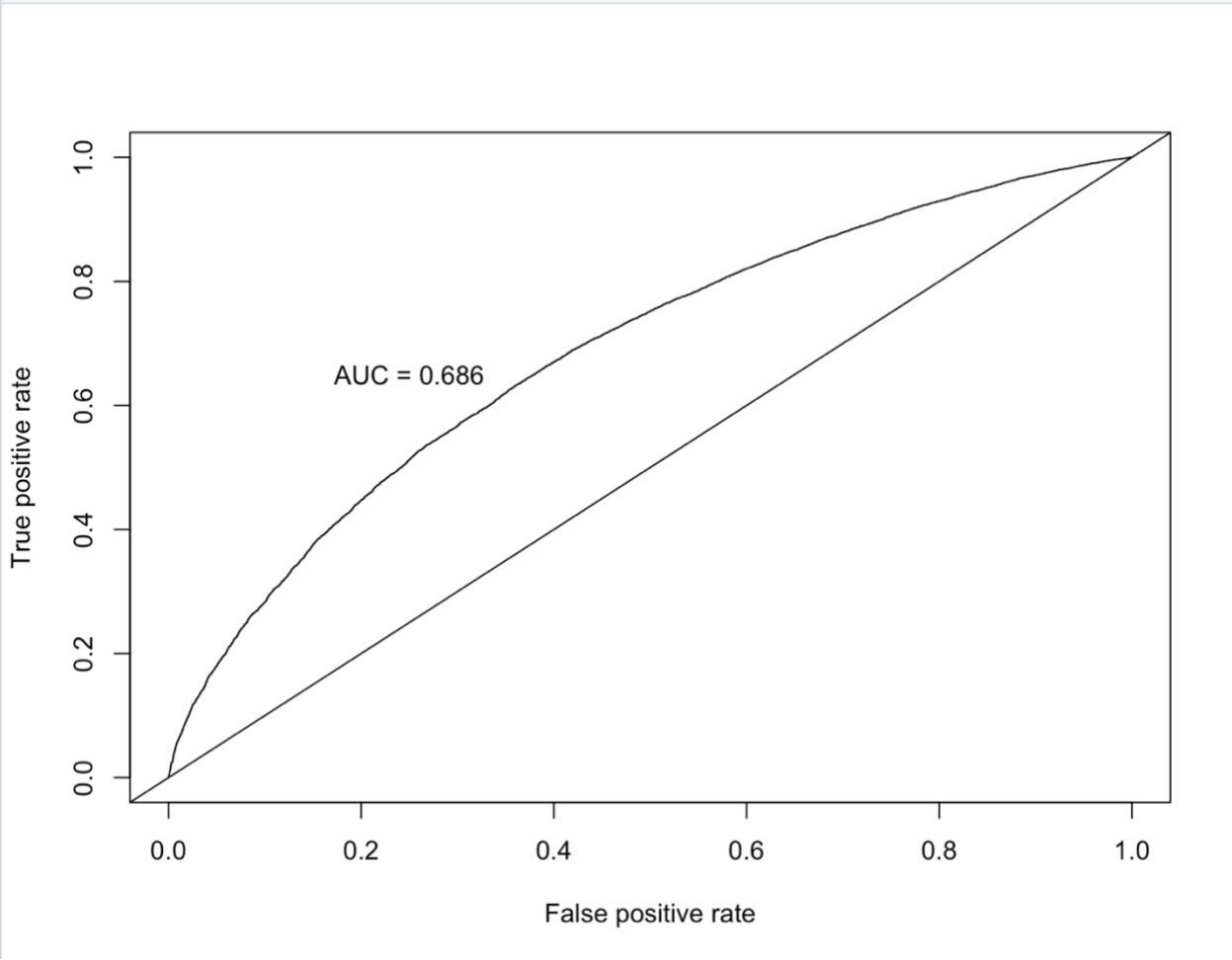
Get the posteriors as a dataframe.

```
loan_raw.lda.predict.posteriors <- as.data.frame(loan_raw.lda.predict$posterior)
# install.packages("ROCR",
lib="/Library/Frameworks/R.framework/Versions/3.5/Resources/library")
library(ROCR)

#create ROC/AUC curve
#prediction function is used for posterier data and test data$diagnosis
pred <- prediction(loan_raw.lda.predict.posteriors[,2], test_raw.df$loan_status)
#pred has various parameters in the list which are predictions, labels, cutoffs, fp,tp,
#tn, fn, n.pos, n.neg, n.neg.pred,n.pos

roc.perf = performance(pred, measure = "tpr", x.measure = "fpr")
auc.train <- performance(pred, measure = "auc")
auc.train <- auc.train@y.values
plot(roc.perf)
```



text(x = .25, y = .65 ,paste("AUC = ", round(auc.train[[1]],3), sep = "")) #the area under the curve is being calculated and displayed which shows that our model has about 98.3% accuracy of predicting

#Ans- Conclusion:- The area under the curve gives us precision of 0.698~ 69.8%.
This implies that the chosen model will be able to predict the correct values 70 times out of 100.



attach(loan)
r <- lda(formula = loan_status ~ ., data = loan) #Species is a categorical variable which is taken against all the other variables

r$counts #what the membership is

```
> r$counts #what the membership is
     0      1
 39579 102722
```

r$means #what the means are

```
> r$means #what the means are
  loan_amnt funded_amnt installment int_rate  issue_d    grade   purpose      dti emp_length home_ownership
0  15937.50    15937.50    489.6917 15.91910 2016.265 4.603300 0.2288840 21.12483   6.288234      0.5754819
1  14022.63    14022.63    433.7595 12.89512 2016.210 5.341154 0.2319951 18.99656   6.401375      0.6635580
  annual_inc      term
0   78381.65 0.6593395
1   84712.98 0.8114912
```

r$scaling #to see the dist formula to get multi linearsity out
r$prior
r$lev

r$svd

```
> r$scaling #to see the dist formula to get multi linearsity out
                        LD1
loan_amnt       4.120070e-06
funded_amnt     4.120070e-06
installment    -9.020776e-04
int_rate       -8.856004e-02
issue_d        -8.101570e-02
grade           2.581637e-01
purpose        -8.084379e-02
dti            -1.399646e-02
emp_length      3.636656e-03
home_ownership  5.717753e-01
annual_inc      9.457092e-07
term            6.178314e-01
> r$prior
        0         1
0.2781358 0.7218642
> r$lev
[1] "0" "1"
> r$svd
[1] 114.5497
```

?lda
r$N
r$call
(prop = r$svd^2/sum(r$svd^2)) #answer=1
r2 <- lda(formula = loan_status ~ ., data = loan, CV = TRUE)

r2 #this will tell us the accuracy and the groups

```
> r2 #this will tell us the accuracy and the groups
$class
   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1
  [56] 1 1 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
 [111] 0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1
 [166] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
 [221] 1 1 0 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
 [276] 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
 [331] 1 0 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 1 1 0 1
 [386] 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
 [441] 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1
 [496] 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 0 1 0 1 0 1 0 0
 [551] 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1
 [606] 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1 1 1 1 1 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1
 [661] 0 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 1 1 0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [716] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 0 0 1 1 1
 [771] 1 1 1 1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
 [826] 1 1 1 1 0 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 0 1 1 1 0 1 1 1
 [881] 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1
 [936] 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0
 [991] 1 1 1 1 1 0 1 1 1 1
 [ reached getOption("max.print") -- omitted 141301 entries ]
Levels: 0 1

$posterior
               0          1
1      0.48680565 0.51319435
3      0.13531768 0.86468232
4      0.20313053 0.79686947
8      0.28491551 0.71508449
9      0.30715014 0.69284986
11     0.49833998 0.50166002
14     0.11084393 0.88915607
18     0.25312475 0.74687525
```

```
              loan_amnt funded_amnt installment int_rate issue_d grade purpose dti emp_length home_ownership
loan_status           0           0           0        0       0     0       0   0          0              0
loan_amnt             1           0           0        0       0     0       0   0          0              0
funded_amnt           0           1           0        0       0     0       0   0          0              0
installment           0           0           1        0       0     0       0   0          0              0
int_rate              0           0           0        1       0     0       0   0          0              0
issue_d               0           0           0        0       1     0       0   0          0              0
grade                 0           0           0        0       0     1       0   0          0              0
purpose               0           0           0        0       0     0       1   0          0              0
dti                   0           0           0        0       0     0       0   1          0              0
emp_length            0           0           0        0       0     0       0   0          1              0
home_ownership        0           0           0        0       0     0       0   0          0              1
annual_inc            0           0           0        0       0     0       0   0          0              0
term                  0           0           0        0       0     0       0   0          0              0
              annual_inc term
loan_status             0    0
loan_amnt               0    0
funded_amnt             0    0
installment             0    0
int_rate                0    0
issue_d                 0    0
grade                   0    0
purpose                 0    0
dti                     0    0
emp_length              0    0
home_ownership          0    0
annual_inc              1    0
term                    0    1
attr(,"term.labels")
 [1] "loan_amnt"      "funded_amnt"    "installment"    "int_rate"       "issue_d"        "grade"
 [7] "purpose"        "dti"            "emp_length"     "home_ownership" "annual_inc"     "term"
attr(,"order")
 [1] 1 1 1 1 1 1 1 1 1 1 1 1
attr(,"intercept")
[1] 1
attr(,"response")
[1] 1
attr(,".Environment")
<environment: R_GlobalEnv>
attr(,"predvars")
list(loan_status, loan_amnt, funded_amnt, installment, int_rate,
    issue_d, grade, purpose, dti, emp_length, home_ownership,
    annual_inc, term)
attr(,"dataClasses")
   loan_status      loan_amnt    funded_amnt     installment       int_rate        issue_d          grade
     "numeric"      "numeric"      "numeric"      "numeric"      "numeric"      "numeric"      "numeric"
       purpose            dti     emp_length home_ownership     annual_inc           term
     "numeric"      "numeric"      "numeric"      "numeric"      "numeric"      "numeric"

$call
lda(formula = loan_status ~ ., data = loan, CV = TRUE)
```

head(r2$class)

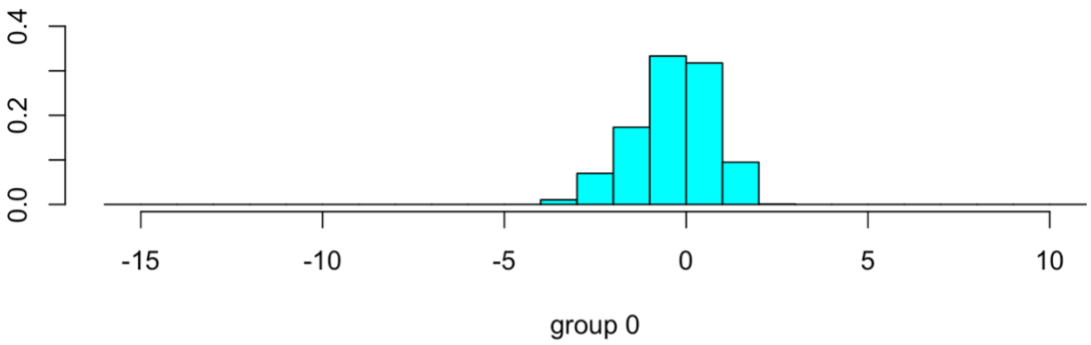
```
> head(r2$class)
[1] 1 1 1 1 1 1
Levels: 0 1
```

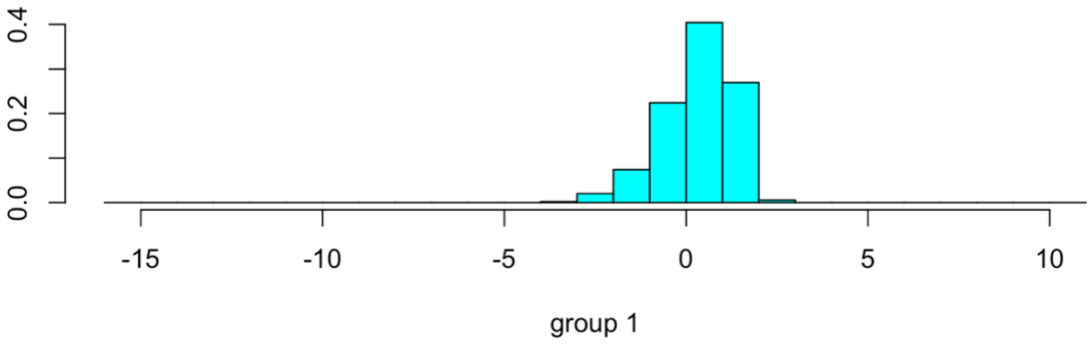
#the Maximum a Posteriori Probability (MAP) classification (a factor)
#posterior: posterior probabilities for the classes.

head(r2$posterior, 3)

```
> head(r2$posterior, 3)
          0         1
1 0.4868056 0.5131944
3 0.1353177 0.8646823
4 0.2031305 0.7968695
```

```
#calculating for the 75% of the overall observations
train <- sample(1:130718, 98038)
r3 <- lda(loan_status ~ ., # training model
       loan,
       #prior = c(0.5,0.5,0.5)/3, #telling the comp use that to adjust the probabilities
       #we are setting 1,1,1 but you can set it by yourself based on the cut-off values
       subset = train)
plda = predict(object = r3, # predictions
          newdata = loan[-train, ])
head(plda$class)
head(plda$posterior, 6) # posterior prob.
head(plda$x, 3)
> head(plda$class)
[1] 1 1 1 0 1 1
Levels: 0 1
> head(plda$posterior, 6) # posterior prob.
           0         1
8  0.2848278 0.7151722
20 0.3022547 0.6977453
50 0.4223669 0.5776331
56 0.5408600 0.4591400
65 0.1775736 0.8224264
68 0.2910572 0.7089428
> head(plda$x, 3)
          LD1
8  -0.1721976
20 -0.2966011
50 -1.0714280

plot(r)
plot(r3) #this function performs better than r
```






```
r <- lda(loan_status ~ .,
        loan)
        #prior = c(1,1,1)/3)
prop.lda = r$svd^2/sum(r$svd^2)
plda <- predict(object = r,
              newdata = loan)
dataset = data.frame(loan_status = loan[,"loan_status"],lda = plda$x)
library(ggplot2)
install.packages("memisc")
library(memisc)
```

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

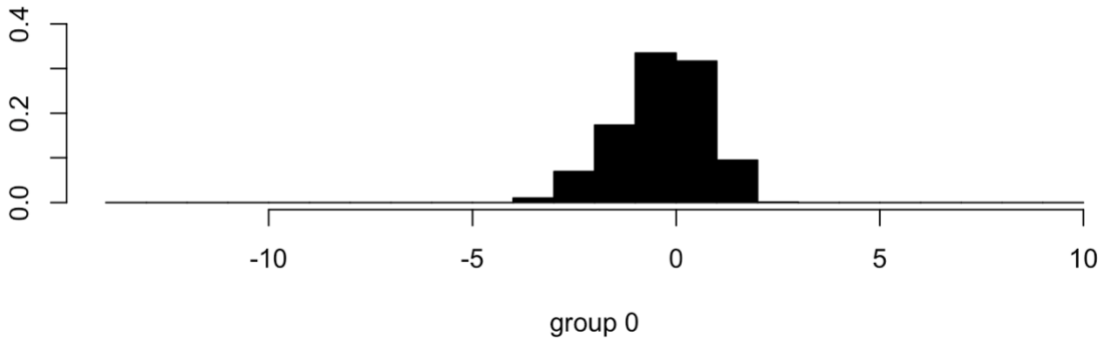
library(dplyr) # has a sample function
set.seed(101) # Nothing is random!!
sample_n(loan,10)

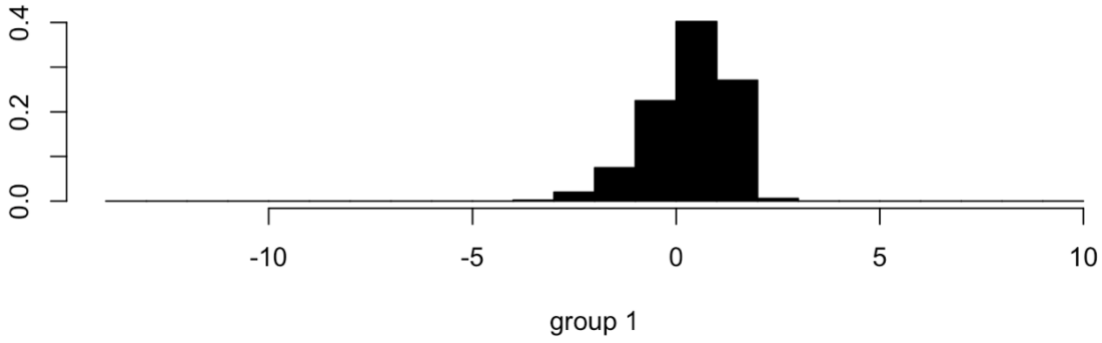
```
> sample_n(loan,10)
   loan_status loan_amnt funded_amnt installment int_rate issue_d grade purpose   dti emp_length home_ownership
1            0     12000       12000      365.67     6.11    2018     7       0  5.48          3              1
2            1     24000       24000      558.32    13.99    2016     5       0 30.10          7              1
3            1     24000       24000      756.40     8.39    2016     6       0 17.78          8              1
4            0     13500       13500      335.44    16.99    2016     4       0 29.40         10              1
5            0     17000       17000      577.56    13.58    2018     5       0  8.10          4              0
6            1     10900       10900      433.33    24.99    2016     3       1 17.07          1              0
7            1     28425       28425      954.88    12.79    2016     5       0 23.09         10              0
8            1      4800        4800      164.03    13.99    2016     5       0 27.07          5              0
9            1     25000       25000      543.44    10.99    2016     6       0 15.21         10              1
10           0     20000       20000      450.88    12.59    2015     5       0 22.55          2              1
   annual_inc term
1      120000    1
2      150000    0
3       97000    1
4       60000    0
5       56000    1
6       50000    1
7      226000    1
8      106000    1
9      115000    0
10      67000    0
```

# Lets take a sample of 75/25 like before. Dplyr preserves class.
training_sample <- sample(c(TRUE, FALSE), nrow(loan), replace = T, prob = c(0.75,0.25)) #no things in two dataset is done by replace
train <- loan[training_sample,]
test <- loan[!training_sample,]

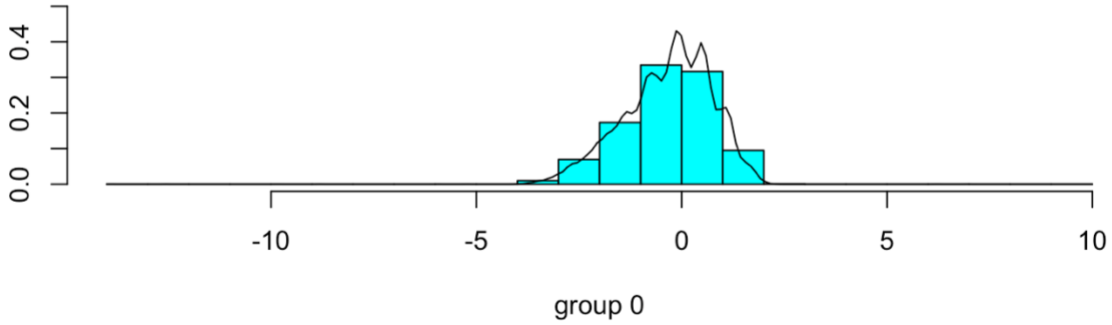
#lets run LDA like before
lda.loan<- lda(loan_status ~ ., train)

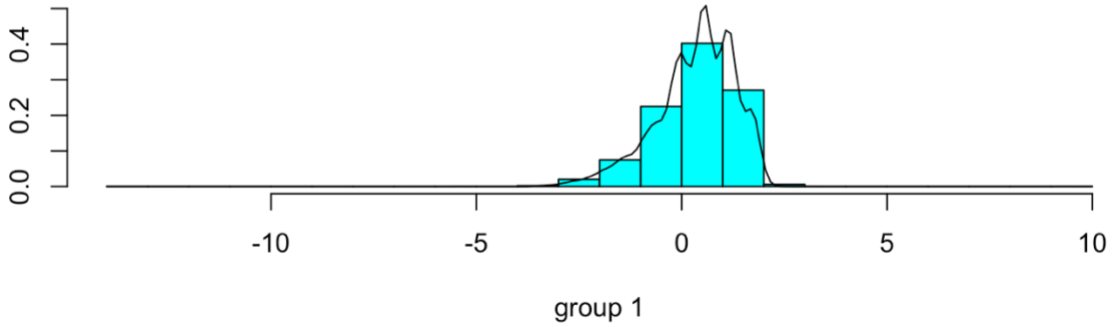
# do a quick plot to understand how good the model is
plot(lda.loan, col = as.integer(train$loan_status))






```
# Sometime bell curves are better
plot(lda.loan, dimen = 1, type = "b")
```





# THis plot shows the essense of LDA. It puts everything on a line and finds cutoffs.
# Partition plots
install.packages("klaR")
library(klaR)
attach(train)
str(train)

#when I do structure on the train dataframe, I can find out that there
#are no factors in the dataset, To make sure that you run partimat variable below,
# the variables of the dataset needs ro be a factor so this line of code won't work in our case
partimat(loan_status ~ purpose+dti+home_ownership+term, data=train, method="lda")

```
# focus on accuracy. Table function
lda.train <- predict(lda.loan)
train$lda <- lda.train$class
table(train$lda,train$loan_status)
```

```
      0     1
0  5021  4460
1 24541 72395
```

```
# 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.loan,test)
test$lda <- lda.test$class
table(test$lda,test$loan_status)
```

```
      0     1
0  1724  1472
1  8293 24395
```