M6L4\_Assignment

Xueyi Fan

June 30, 2016

# Assignment:

1. Go to the UC Irvine Machine Learning Repository and find a dataset for supervised classification. Every student MUST use a different dataset so you MUST get approved for which you can going to use. This can be the same dataset you used for the unsupervised clustering as long as the data has some labeled data.
2. Classify your data using Linear Discriminant Analysis (LDA). Answer the following questions:

* Does the number of predictor variables for LDA make a difference? Try for a range of models using differing numbers of predictor variables.
* What determines the number of linear discriminants in LDA.
* Does scaling, normalization or leaving the data unscaled make a difference for LDA?

library("ggplot2")  
library("MASS")  
library("car")

# Answer:

## Loading my data

Here, I choose the [Breast Cancer Wisconsin (Diagnostic) data set](https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/)

data\_url <- 'https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data'  
  
data <- read.table(url(data\_url), sep = ',')  
  
names(data) <- c('ID number', 'Diagnosis','radius\_mean','texure\_mean','perimeter\_mean','area\_mean','smoothness\_mean','compactness\_mean','concavity\_mean','concave\_points\_mean','symmetry\_mean','fractal\_dimension\_mean', 'radius\_SE','texure\_SE','perimeter\_SE','area\_SE','smoothness\_SE','compactness\_SE','concavity\_SE','concave\_points\_SE','symmetry\_SE','fractal\_dimension\_SE','radius\_worst','texure\_worst','perimeter\_worst','area\_worst','smoothness\_worst','compactness\_worst','concavity\_worst','concave\_points\_worst','symmetry\_worst','fractal\_dimension\_worst')   
  
head(data)

## ID number Diagnosis radius\_mean texure\_mean perimeter\_mean area\_mean  
## 1 842302 M 17.99 10.38 122.80 1001.0  
## 2 842517 M 20.57 17.77 132.90 1326.0  
## 3 84300903 M 19.69 21.25 130.00 1203.0  
## 4 84348301 M 11.42 20.38 77.58 386.1  
## 5 84358402 M 20.29 14.34 135.10 1297.0  
## 6 843786 M 12.45 15.70 82.57 477.1  
## smoothness\_mean compactness\_mean concavity\_mean concave\_points\_mean  
## 1 0.11840 0.27760 0.3001 0.14710  
## 2 0.08474 0.07864 0.0869 0.07017  
## 3 0.10960 0.15990 0.1974 0.12790  
## 4 0.14250 0.28390 0.2414 0.10520  
## 5 0.10030 0.13280 0.1980 0.10430  
## 6 0.12780 0.17000 0.1578 0.08089  
## symmetry\_mean fractal\_dimension\_mean radius\_SE texure\_SE perimeter\_SE  
## 1 0.2419 0.07871 1.0950 0.9053 8.589  
## 2 0.1812 0.05667 0.5435 0.7339 3.398  
## 3 0.2069 0.05999 0.7456 0.7869 4.585  
## 4 0.2597 0.09744 0.4956 1.1560 3.445  
## 5 0.1809 0.05883 0.7572 0.7813 5.438  
## 6 0.2087 0.07613 0.3345 0.8902 2.217  
## area\_SE smoothness\_SE compactness\_SE concavity\_SE concave\_points\_SE  
## 1 153.40 0.006399 0.04904 0.05373 0.01587  
## 2 74.08 0.005225 0.01308 0.01860 0.01340  
## 3 94.03 0.006150 0.04006 0.03832 0.02058  
## 4 27.23 0.009110 0.07458 0.05661 0.01867  
## 5 94.44 0.011490 0.02461 0.05688 0.01885  
## 6 27.19 0.007510 0.03345 0.03672 0.01137  
## symmetry\_SE fractal\_dimension\_SE radius\_worst texure\_worst  
## 1 0.03003 0.006193 25.38 17.33  
## 2 0.01389 0.003532 24.99 23.41  
## 3 0.02250 0.004571 23.57 25.53  
## 4 0.05963 0.009208 14.91 26.50  
## 5 0.01756 0.005115 22.54 16.67  
## 6 0.02165 0.005082 15.47 23.75  
## perimeter\_worst area\_worst smoothness\_worst compactness\_worst  
## 1 184.60 2019.0 0.1622 0.6656  
## 2 158.80 1956.0 0.1238 0.1866  
## 3 152.50 1709.0 0.1444 0.4245  
## 4 98.87 567.7 0.2098 0.8663  
## 5 152.20 1575.0 0.1374 0.2050  
## 6 103.40 741.6 0.1791 0.5249  
## concavity\_worst concave\_points\_worst symmetry\_worst  
## 1 0.7119 0.2654 0.4601  
## 2 0.2416 0.1860 0.2750  
## 3 0.4504 0.2430 0.3613  
## 4 0.6869 0.2575 0.6638  
## 5 0.4000 0.1625 0.2364  
## 6 0.5355 0.1741 0.3985  
## fractal\_dimension\_worst  
## 1 0.11890  
## 2 0.08902  
## 3 0.08758  
## 4 0.17300  
## 5 0.07678  
## 6 0.12440

str(data)

## 'data.frame': 569 obs. of 32 variables:  
## $ ID number : int 842302 842517 84300903 84348301 84358402 843786 844359 84458202 844981 84501001 ...  
## $ Diagnosis : Factor w/ 2 levels "B","M": 2 2 2 2 2 2 2 2 2 2 ...  
## $ radius\_mean : num 18 20.6 19.7 11.4 20.3 ...  
## $ texure\_mean : num 10.4 17.8 21.2 20.4 14.3 ...  
## $ perimeter\_mean : num 122.8 132.9 130 77.6 135.1 ...  
## $ area\_mean : num 1001 1326 1203 386 1297 ...  
## $ smoothness\_mean : num 0.1184 0.0847 0.1096 0.1425 0.1003 ...  
## $ compactness\_mean : num 0.2776 0.0786 0.1599 0.2839 0.1328 ...  
## $ concavity\_mean : num 0.3001 0.0869 0.1974 0.2414 0.198 ...  
## $ concave\_points\_mean : num 0.1471 0.0702 0.1279 0.1052 0.1043 ...  
## $ symmetry\_mean : num 0.242 0.181 0.207 0.26 0.181 ...  
## $ fractal\_dimension\_mean : num 0.0787 0.0567 0.06 0.0974 0.0588 ...  
## $ radius\_SE : num 1.095 0.543 0.746 0.496 0.757 ...  
## $ texure\_SE : num 0.905 0.734 0.787 1.156 0.781 ...  
## $ perimeter\_SE : num 8.59 3.4 4.58 3.44 5.44 ...  
## $ area\_SE : num 153.4 74.1 94 27.2 94.4 ...  
## $ smoothness\_SE : num 0.0064 0.00522 0.00615 0.00911 0.01149 ...  
## $ compactness\_SE : num 0.049 0.0131 0.0401 0.0746 0.0246 ...  
## $ concavity\_SE : num 0.0537 0.0186 0.0383 0.0566 0.0569 ...  
## $ concave\_points\_SE : num 0.0159 0.0134 0.0206 0.0187 0.0188 ...  
## $ symmetry\_SE : num 0.03 0.0139 0.0225 0.0596 0.0176 ...  
## $ fractal\_dimension\_SE : num 0.00619 0.00353 0.00457 0.00921 0.00511 ...  
## $ radius\_worst : num 25.4 25 23.6 14.9 22.5 ...  
## $ texure\_worst : num 17.3 23.4 25.5 26.5 16.7 ...  
## $ perimeter\_worst : num 184.6 158.8 152.5 98.9 152.2 ...  
## $ area\_worst : num 2019 1956 1709 568 1575 ...  
## $ smoothness\_worst : num 0.162 0.124 0.144 0.21 0.137 ...  
## $ compactness\_worst : num 0.666 0.187 0.424 0.866 0.205 ...  
## $ concavity\_worst : num 0.712 0.242 0.45 0.687 0.4 ...  
## $ concave\_points\_worst : num 0.265 0.186 0.243 0.258 0.163 ...  
## $ symmetry\_worst : num 0.46 0.275 0.361 0.664 0.236 ...  
## $ fractal\_dimension\_worst: num 0.1189 0.089 0.0876 0.173 0.0768 ...

summary(data)

## ID number Diagnosis radius\_mean texure\_mean   
## Min. : 8670 B:357 Min. : 6.981 Min. : 9.71   
## 1st Qu.: 869218 M:212 1st Qu.:11.700 1st Qu.:16.17   
## Median : 906024 Median :13.370 Median :18.84   
## Mean : 30371831 Mean :14.127 Mean :19.29   
## 3rd Qu.: 8813129 3rd Qu.:15.780 3rd Qu.:21.80   
## Max. :911320502 Max. :28.110 Max. :39.28   
## perimeter\_mean area\_mean smoothness\_mean compactness\_mean   
## Min. : 43.79 Min. : 143.5 Min. :0.05263 Min. :0.01938   
## 1st Qu.: 75.17 1st Qu.: 420.3 1st Qu.:0.08637 1st Qu.:0.06492   
## Median : 86.24 Median : 551.1 Median :0.09587 Median :0.09263   
## Mean : 91.97 Mean : 654.9 Mean :0.09636 Mean :0.10434   
## 3rd Qu.:104.10 3rd Qu.: 782.7 3rd Qu.:0.10530 3rd Qu.:0.13040   
## Max. :188.50 Max. :2501.0 Max. :0.16340 Max. :0.34540   
## concavity\_mean concave\_points\_mean symmetry\_mean   
## Min. :0.00000 Min. :0.00000 Min. :0.1060   
## 1st Qu.:0.02956 1st Qu.:0.02031 1st Qu.:0.1619   
## Median :0.06154 Median :0.03350 Median :0.1792   
## Mean :0.08880 Mean :0.04892 Mean :0.1812   
## 3rd Qu.:0.13070 3rd Qu.:0.07400 3rd Qu.:0.1957   
## Max. :0.42680 Max. :0.20120 Max. :0.3040   
## fractal\_dimension\_mean radius\_SE texure\_SE perimeter\_SE   
## Min. :0.04996 Min. :0.1115 Min. :0.3602 Min. : 0.757   
## 1st Qu.:0.05770 1st Qu.:0.2324 1st Qu.:0.8339 1st Qu.: 1.606   
## Median :0.06154 Median :0.3242 Median :1.1080 Median : 2.287   
## Mean :0.06280 Mean :0.4052 Mean :1.2169 Mean : 2.866   
## 3rd Qu.:0.06612 3rd Qu.:0.4789 3rd Qu.:1.4740 3rd Qu.: 3.357   
## Max. :0.09744 Max. :2.8730 Max. :4.8850 Max. :21.980   
## area\_SE smoothness\_SE compactness\_SE concavity\_SE   
## Min. : 6.802 Min. :0.001713 Min. :0.002252 Min. :0.00000   
## 1st Qu.: 17.850 1st Qu.:0.005169 1st Qu.:0.013080 1st Qu.:0.01509   
## Median : 24.530 Median :0.006380 Median :0.020450 Median :0.02589   
## Mean : 40.337 Mean :0.007041 Mean :0.025478 Mean :0.03189   
## 3rd Qu.: 45.190 3rd Qu.:0.008146 3rd Qu.:0.032450 3rd Qu.:0.04205   
## Max. :542.200 Max. :0.031130 Max. :0.135400 Max. :0.39600   
## concave\_points\_SE symmetry\_SE fractal\_dimension\_SE  
## Min. :0.000000 Min. :0.007882 Min. :0.0008948   
## 1st Qu.:0.007638 1st Qu.:0.015160 1st Qu.:0.0022480   
## Median :0.010930 Median :0.018730 Median :0.0031870   
## Mean :0.011796 Mean :0.020542 Mean :0.0037949   
## 3rd Qu.:0.014710 3rd Qu.:0.023480 3rd Qu.:0.0045580   
## Max. :0.052790 Max. :0.078950 Max. :0.0298400   
## radius\_worst texure\_worst perimeter\_worst area\_worst   
## Min. : 7.93 Min. :12.02 Min. : 50.41 Min. : 185.2   
## 1st Qu.:13.01 1st Qu.:21.08 1st Qu.: 84.11 1st Qu.: 515.3   
## Median :14.97 Median :25.41 Median : 97.66 Median : 686.5   
## Mean :16.27 Mean :25.68 Mean :107.26 Mean : 880.6   
## 3rd Qu.:18.79 3rd Qu.:29.72 3rd Qu.:125.40 3rd Qu.:1084.0   
## Max. :36.04 Max. :49.54 Max. :251.20 Max. :4254.0   
## smoothness\_worst compactness\_worst concavity\_worst concave\_points\_worst  
## Min. :0.07117 Min. :0.02729 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.11660 1st Qu.:0.14720 1st Qu.:0.1145 1st Qu.:0.06493   
## Median :0.13130 Median :0.21190 Median :0.2267 Median :0.09993   
## Mean :0.13237 Mean :0.25427 Mean :0.2722 Mean :0.11461   
## 3rd Qu.:0.14600 3rd Qu.:0.33910 3rd Qu.:0.3829 3rd Qu.:0.16140   
## Max. :0.22260 Max. :1.05800 Max. :1.2520 Max. :0.29100   
## symmetry\_worst fractal\_dimension\_worst  
## Min. :0.1565 Min. :0.05504   
## 1st Qu.:0.2504 1st Qu.:0.07146   
## Median :0.2822 Median :0.08004   
## Mean :0.2901 Mean :0.08395   
## 3rd Qu.:0.3179 3rd Qu.:0.09208   
## Max. :0.6638 Max. :0.20750

#shuffle the data   
set.seed(123)  
cancer\_data <- data[order(runif(nrow(data))),]  
cancer\_data[,2]

## [1] B M B B B B B B B M B B B B B B M M B M B B B B B B B M B B B B B M B  
## [36] B M B B B B M B B B B B B B B B B M M B B B B B B B B M B M B B B B B  
## [71] M B B M B B B B M M M B M B B B B M B B B M B B B B B B M B B M B B M  
## [106] M M B B B M B B M B B B B B M M B M M B B B B M B B B M B M B B B B M  
## [141] B B B M B B M B B M B B M B B M B B B M B B B M M M M B B B B B B M B  
## [176] B B B M B M B B M M M B M B B B M M B B M M M B B B M M M B B M B B M  
## [211] B B B M B M M B B B B B B B M B B B B B B B M B B B M B B B M M M M B  
## [246] M M M M M M B M B M B B B M M B B B B B B M B B M M M M M B B B M M B  
## [281] B M B M B B B B M B M B B B B M B B B B B B M M M M B M B B M B M B M  
## [316] B B B B B B M B B B B M B M B B B M M M B M B M B B M B M M M B M M B  
## [351] M M M B M B M B B B B B B B B M M B B B B M M M M M B M M B B B B B B  
## [386] B B M M M M M M B B M M B B M M B B B B M B M M B M M B M B B B M B B  
## [421] B B B B M B B M B B B B M B M B M M M B B M M M B M B M M M M B M M M  
## [456] B M B B B B B M B B B M B B M B M B M B B B B M B B M B B M B B B B B  
## [491] B M B M M M M B B M B M M B B B B M B M M M B B M B M B B B M B M M M  
## [526] M B B B M B M B B M B M M B B M B M M B B B M B M B B M B M M B B B M  
## [561] B B M B B B M M B  
## Levels: B M

## Classify my data using Linear Discriminant Analysis (LDA)

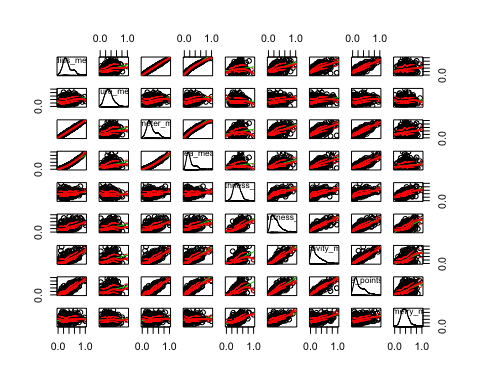
#to normalize data  
normalize <- function(x){  
 return((x-min(x))/(max(x)-min(x)))  
}  
cancer.normalized <- as.data.frame(lapply(cancer\_data[,3:32],normalize))  
head(cancer.normalized)

## radius\_mean texure\_mean perimeter\_mean area\_mean smoothness\_mean  
## 1 0.32746462 0.2336828 0.31221063 0.19338282 0.1412837  
## 2 0.32273179 0.2056138 0.32229977 0.18689290 0.4339623  
## 3 0.20393772 0.1126141 0.19653099 0.10371156 0.4126569  
## 4 0.09555587 0.1586067 0.08686338 0.04360551 0.1572628  
## 5 0.25931185 0.3442678 0.25319605 0.13904560 0.2878036  
## 6 0.21056368 0.2570172 0.20641283 0.10795334 0.5106978  
## compactness\_mean concavity\_mean concave\_points\_mean symmetry\_mean  
## 1 0.10370529 0.052108716 0.06655070 0.3803030  
## 2 0.33316974 0.182497657 0.25193837 0.3040404  
## 3 0.17391571 0.076499531 0.13692843 0.3580808  
## 4 0.03613275 0.008624649 0.01725646 0.3676768  
## 5 0.25157966 0.160028116 0.15402584 0.3641414  
## 6 0.23151954 0.047586692 0.09249503 0.2954545  
## fractal\_dimension\_mean radius\_SE texure\_SE perimeter\_SE area\_SE  
## 1 0.1137321 0.01593337 0.04773692 0.02992037 0.013500237  
## 2 0.3306655 0.06054680 0.05752740 0.05654243 0.030907848  
## 3 0.2683235 0.02857143 0.03737182 0.01917731 0.011893956  
## 4 0.3868997 0.02284990 0.20880481 0.01823493 0.005579027  
## 5 0.2639006 0.09150824 0.15023868 0.08585026 0.040190662  
## 6 0.3298231 0.06257469 0.34207037 0.05560006 0.027228342  
## smoothness\_SE compactness\_SE concavity\_SE concave\_points\_SE symmetry\_SE  
## 1 0.05422035 0.08184877 0.025010101 0.09153249 0.07426690  
## 2 0.10211782 0.13817707 0.044419192 0.17438909 0.06075871  
## 3 0.16177720 0.06735362 0.032373737 0.16762644 0.12717397  
## 4 0.19294965 0.01975997 0.009295455 0.06577003 0.26915067  
## 5 0.16374885 0.23611320 0.118989899 0.26577003 0.27463837  
## 6 0.16830404 0.11339262 0.029242424 0.18033719 0.21019305  
## fractal\_dimension\_SE radius\_worst texure\_worst perimeter\_worst  
## 1 0.04146456 0.25649235 0.2606610 0.25295084  
## 2 0.07722178 0.30736393 0.2356077 0.29827183  
## 3 0.06640825 0.15617218 0.1108742 0.13875193  
## 4 0.04346835 0.06293134 0.2145522 0.05224364  
## 5 0.12976936 0.19672714 0.2945096 0.18785796  
## 6 0.09090281 0.17253646 0.3928571 0.16061557  
## area\_worst smoothness\_worst compactness\_worst concavity\_worst  
## 1 0.13116890 0.1497061 0.16814623 0.11054313  
## 2 0.15414864 0.4617975 0.31716972 0.22196486  
## 3 0.06692391 0.4267979 0.11973300 0.10183706  
## 4 0.02465100 0.1812058 0.02428423 0.01175719  
## 5 0.08997739 0.2234036 0.18289335 0.15207668  
## 6 0.07810657 0.4631183 0.14738384 0.06156550  
## concave\_points\_worst symmetry\_worst fractal\_dimension\_worst  
## 1 0.21381443 0.2195939 0.1439066  
## 2 0.47525773 0.2018529 0.3145743  
## 3 0.30068729 0.2302385 0.1651581  
## 4 0.04773196 0.2810960 0.1508592  
## 5 0.26202749 0.2207767 0.1482356  
## 6 0.22000000 0.2008673 0.1700118

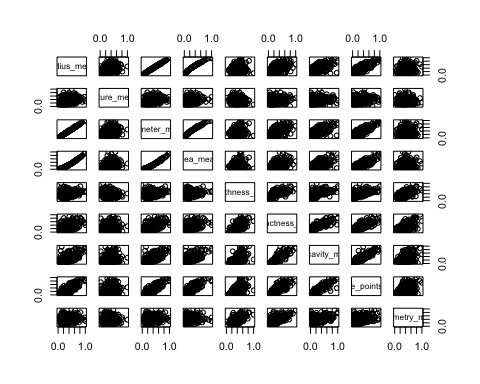
summary(cancer.normalized)

## radius\_mean texure\_mean perimeter\_mean area\_mean   
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.2233 1st Qu.:0.2185 1st Qu.:0.2168 1st Qu.:0.1174   
## Median :0.3024 Median :0.3088 Median :0.2933 Median :0.1729   
## Mean :0.3382 Mean :0.3240 Mean :0.3329 Mean :0.2169   
## 3rd Qu.:0.4164 3rd Qu.:0.4089 3rd Qu.:0.4168 3rd Qu.:0.2711   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## smoothness\_mean compactness\_mean concavity\_mean concave\_points\_mean  
## Min. :0.0000 Min. :0.0000 Min. :0.00000 Min. :0.0000   
## 1st Qu.:0.3046 1st Qu.:0.1397 1st Qu.:0.06926 1st Qu.:0.1009   
## Median :0.3904 Median :0.2247 Median :0.14419 Median :0.1665   
## Mean :0.3948 Mean :0.2606 Mean :0.20806 Mean :0.2431   
## 3rd Qu.:0.4755 3rd Qu.:0.3405 3rd Qu.:0.30623 3rd Qu.:0.3678   
## Max. :1.0000 Max. :1.0000 Max. :1.00000 Max. :1.0000   
## symmetry\_mean fractal\_dimension\_mean radius\_SE   
## Min. :0.0000 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.2823 1st Qu.:0.1630 1st Qu.:0.04378   
## Median :0.3697 Median :0.2439 Median :0.07702   
## Mean :0.3796 Mean :0.2704 Mean :0.10635   
## 3rd Qu.:0.4530 3rd Qu.:0.3404 3rd Qu.:0.13304   
## Max. :1.0000 Max. :1.0000 Max. :1.00000   
## texure\_SE perimeter\_SE area\_SE smoothness\_SE   
## Min. :0.0000 Min. :0.00000 Min. :0.00000 Min. :0.0000   
## 1st Qu.:0.1047 1st Qu.:0.04000 1st Qu.:0.02064 1st Qu.:0.1175   
## Median :0.1653 Median :0.07209 Median :0.03311 Median :0.1586   
## Mean :0.1893 Mean :0.09938 Mean :0.06264 Mean :0.1811   
## 3rd Qu.:0.2462 3rd Qu.:0.12251 3rd Qu.:0.07170 3rd Qu.:0.2187   
## Max. :1.0000 Max. :1.00000 Max. :1.00000 Max. :1.0000   
## compactness\_SE concavity\_SE concave\_points\_SE symmetry\_SE   
## Min. :0.00000 Min. :0.00000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.08132 1st Qu.:0.03811 1st Qu.:0.1447 1st Qu.:0.1024   
## Median :0.13667 Median :0.06538 Median :0.2070 Median :0.1526   
## Mean :0.17444 Mean :0.08054 Mean :0.2235 Mean :0.1781   
## 3rd Qu.:0.22680 3rd Qu.:0.10619 3rd Qu.:0.2787 3rd Qu.:0.2195   
## Max. :1.00000 Max. :1.00000 Max. :1.0000 Max. :1.0000   
## fractal\_dimension\_SE radius\_worst texure\_worst perimeter\_worst   
## Min. :0.00000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.04675 1st Qu.:0.1807 1st Qu.:0.2415 1st Qu.:0.1678   
## Median :0.07919 Median :0.2504 Median :0.3569 Median :0.2353   
## Mean :0.10019 Mean :0.2967 Mean :0.3640 Mean :0.2831   
## 3rd Qu.:0.12656 3rd Qu.:0.3863 3rd Qu.:0.4717 3rd Qu.:0.3735   
## Max. :1.00000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## area\_worst smoothness\_worst compactness\_worst concavity\_worst   
## Min. :0.00000 Min. :0.0000 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.08113 1st Qu.:0.3000 1st Qu.:0.1163 1st Qu.:0.09145   
## Median :0.12321 Median :0.3971 Median :0.1791 Median :0.18107   
## Mean :0.17091 Mean :0.4041 Mean :0.2202 Mean :0.21740   
## 3rd Qu.:0.22090 3rd Qu.:0.4942 3rd Qu.:0.3025 3rd Qu.:0.30583   
## Max. :1.00000 Max. :1.0000 Max. :1.0000 Max. :1.00000   
## concave\_points\_worst symmetry\_worst fractal\_dimension\_worst  
## Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.2231 1st Qu.:0.1851 1st Qu.:0.1077   
## Median :0.3434 Median :0.2478 Median :0.1640   
## Mean :0.3938 Mean :0.2633 Mean :0.1896   
## 3rd Qu.:0.5546 3rd Qu.:0.3182 3rd Qu.:0.2429   
## Max. :1.0000 Max. :1.0000 Max. :1.0000

scatterplotMatrix(cancer.normalized[,1:9])



pairs(cancer.normalized[,1:9])



#split my data into training part and testing part  
cancer.train <- cancer.normalized[1:500,]  
cancer.test <- cancer.normalized[501:569,]  
cancer.train.target <- cancer\_data[1:500,2]  
cancer.test.target <- cancer\_data[501:569,2]  
  
#using LDA  
cancer.lda <- lda(cancer.train.target~., data=cancer.train)  
cancer.lda

## Call:  
## lda(cancer.train.target ~ ., data = cancer.train)  
##   
## Prior probabilities of groups:  
## B M   
## 0.636 0.364   
##   
## Group means:  
## radius\_mean texure\_mean perimeter\_mean area\_mean smoothness\_mean  
## B 0.2434860 0.2761872 0.2359636 0.1347901 0.3571072  
## M 0.5055647 0.4008964 0.5039924 0.3632951 0.4498223  
## compactness\_mean concavity\_mean concave\_points\_mean symmetry\_mean  
## B 0.1863355 0.1088961 0.1272673 0.3440347  
## M 0.3850160 0.3779686 0.4438157 0.4381452  
## fractal\_dimension\_mean radius\_SE texure\_SE perimeter\_SE area\_SE  
## B 0.2738512 0.06239135 0.1843959 0.05853974 0.02673766  
## M 0.2617620 0.18464782 0.1898994 0.17324482 0.12762002  
## smoothness\_SE compactness\_SE concavity\_SE concave\_points\_SE symmetry\_SE  
## B 0.1864199 0.1464068 0.06717828 0.1874782 0.1781531  
## M 0.1705850 0.2266939 0.10533217 0.2886295 0.1744843  
## fractal\_dimension\_SE radius\_worst texure\_worst perimeter\_worst  
## B 0.09686564 0.1932006 0.3044808 0.1816268  
## M 0.10850162 0.4777288 0.4589538 0.4604193  
## area\_worst smoothness\_worst compactness\_worst concavity\_worst  
## B 0.09146795 0.3528008 0.1513911 0.1341106  
## M 0.31102725 0.4781504 0.3347875 0.3569361  
## concave\_points\_worst symmetry\_worst fractal\_dimension\_worst  
## B 0.2558656 0.2249001 0.1619571  
## M 0.6317705 0.3269897 0.2332296  
##   
## Coefficients of linear discriminants:  
## LD1  
## radius\_mean -27.0093131  
## texure\_mean 0.5515401  
## perimeter\_mean 20.7218668  
## area\_mean 5.0222555  
## smoothness\_mean 0.5375216  
## compactness\_mean -7.0712882  
## concavity\_mean 2.0040579  
## concave\_points\_mean 2.7833787  
## symmetry\_mean -0.1791856  
## fractal\_dimension\_mean -0.0956335  
## radius\_SE 5.8758980  
## texure\_SE -0.6438832  
## perimeter\_SE -1.1641631  
## area\_SE -3.2373755  
## smoothness\_SE 2.5972188  
## compactness\_SE 0.6546591  
## concavity\_SE -6.9505631  
## concave\_points\_SE 3.4254229  
## symmetry\_SE -0.2616661  
## fractal\_dimension\_SE -1.2111637  
## radius\_worst 29.7682102  
## texure\_worst 1.6442592  
## perimeter\_worst -5.3760345  
## area\_worst -20.4958686  
## smoothness\_worst 0.1633950  
## compactness\_worst -0.5936693  
## concavity\_worst 2.5707552  
## concave\_points\_worst 0.2113588  
## symmetry\_worst 2.6803196  
## fractal\_dimension\_worst 3.8892750

## Does the number of predictor variables for LDA make a difference? Try for a range of models using differing numbers of predictor variables.

#use whole variables(30)  
cancer.whole.var <- cancer.normalized[,1:30]  
  
#split the data  
cancer.whole.train <- cancer.whole.var[1:500,]  
cancer.whole.test <- cancer.whole.var[501:569,]  
cancer.whole.train.target <- cancer\_data[1:500,2]  
cancer.whole.test.target <- cancer\_data[501:569,2]  
  
#using LDA  
cancer.whole.lda <- lda(cancer.whole.train.target~., data=cancer.whole.train)  
cancer.whole.lda.pred <- predict(cancer.whole.lda, newdata = cancer.whole.test)  
whole.cm <- table(cancer.whole.lda.pred$class, cancer.whole.test.target)  
  
  
###########################  
#use half of the variables(15)  
cancer.half.var <- cancer.normalized [,1:15]  
  
#split the data  
cancer.half.train <- cancer.half.var[1:500,]  
cancer.half.test <- cancer.half.var[501:569,]  
cancer.half.train.target <- cancer\_data[1:500,2]  
cancer.half.test.target <- cancer\_data[501:569,2]  
  
#using LDA  
cancer.half.lda <- lda(cancer.half.train.target~., data=cancer.half.train)  
cancer.half.lda.pred <- predict(cancer.half.lda, newdata = cancer.half.test)  
half.cm <- table(cancer.half.lda.pred$class, cancer.half.test.target)  
  
###########################  
#use ten variables  
cancer.ten.var <- cancer.normalized[,1:10]  
  
#split the data  
cancer.ten.train <- cancer.ten.var[1:500,]  
cancer.ten.test <- cancer.ten.var[501:569,]  
cancer.ten.train.target <- cancer\_data[1:500,2]  
cancer.ten.test.target <- cancer\_data[501:569,2]  
  
#using LDA  
cancer.ten.lda <- lda(cancer.ten.train.target~., data=cancer.ten.train)  
cancer.ten.lda.pred <- predict(cancer.ten.lda, newdata = cancer.ten.test)  
ten.cm <- table(cancer.ten.lda.pred$class, cancer.ten.test.target)  
  
###########################  
#use five variables  
cancer.five.var<- cancer.normalized[,1:5]  
  
#split the data  
cancer.five.train <- cancer.five.var[1:500,]  
cancer.five.test <- cancer.five.var[501:569,]  
cancer.five.train.target <- cancer\_data[1:500,2]  
cancer.five.test.target <- cancer\_data[501:569,2]  
  
#using LDA  
cancer.five.lda <- lda(cancer.five.train.target~., data=cancer.five.train)  
cancer.five.lda.pred <- predict(cancer.five.lda, newdata = cancer.five.test)  
five.cm <- table(cancer.five.lda.pred$class, cancer.five.test.target)  
  
###########################  
#use two variables  
cancer.two.var<- cancer.normalized[,1:2]  
  
#split the data  
cancer.two.train <- cancer.two.var[1:500,]  
cancer.two.test <- cancer.two.var[501:569,]  
cancer.two.train.target <- cancer\_data[1:500,2]  
cancer.two.test.target <- cancer\_data[501:569,2]  
  
#using LDA  
cancer.two.lda <- lda(cancer.two.train.target~., data=cancer.two.train)  
cancer.two.lda.pred <- predict(cancer.two.lda, newdata = cancer.two.test)  
two.cm <- table(cancer.two.lda.pred$class, cancer.two.test.target)  
  
########################  
#comparing models using differing numbers of predictor variables  
whole.cm

## cancer.whole.test.target  
## B M  
## B 39 4  
## M 0 26

half.cm

## cancer.half.test.target  
## B M  
## B 39 6  
## M 0 24

ten.cm

## cancer.ten.test.target  
## B M  
## B 39 5  
## M 0 25

five.cm

## cancer.five.test.target  
## B M  
## B 37 6  
## M 2 24

two.cm

## cancer.two.test.target  
## B M  
## B 38 11  
## M 1 19

When I use all variables, there are four misclassifications. When I use half of the variables, there are six misclassifications. When I use ten variables, there are five misclassificatoins. When I use five variables, there are eight misclassifications. When I use first two variables, there are 12 misclassifications. It is easy to find that different variables will have different perfomance. So the number of predictor variables for LDA makes a difference.

If I want to got the best k of dimensions of the new feature subspae, it need to compute eigenvectors and collect them.

## What determines the number of linear discriminants in LDA.

The number of classes determines the number of linear discriminants in LDA. The LDA will find at most k-1 linear discriminants. In my data set, I have two classes, so it has one linear discriminant.

## Does scaling, normalization or leaving the data unscaled make a difference for LDA?

#normalized data  
normalize <- function(x){  
 return((x-min(x))/(max(x)-min(x)))  
}  
cancer.normalized <- as.data.frame(lapply(cancer\_data[,3:32],normalize))  
  
#split my data into training part and testing part  
cancer.normalized.train <- cancer.normalized[1:500,]  
cancer.normalized.test <- cancer.normalized[501:569,]  
cancer.normalized.train.target <- cancer\_data[1:500,2]  
cancer.normalized.test.target <- cancer\_data[501:569,2]  
  
#LDA using normalized data  
cancer.normalized.lda <- lda(cancer.normalized.train.target~., data=cancer.normalized.train)  
cancer.normalized.lda.pred <- predict(cancer.normalized.lda, newdata = cancer.normalized.test)  
normalized.cm <- table(cancer.normalized.lda.pred$class, cancer.normalized.test.target)  
  
##################  
#scaling data  
cancer.scaled <- as.data.frame(lapply(cancer\_data[,3:32], scale))  
  
#split my data into training part and testing part  
cancer.scaled.train <- cancer.scaled[1:500,]  
cancer.scaled.test <- cancer.scaled[501:569,]  
cancer.scaled.train.target <- cancer\_data[1:500,2]  
cancer.scaled.test.target <- cancer\_data[501:569,2]  
  
#LDA using scaling data  
cancer.scaled.lda <- lda(cancer.scaled.train.target~., data=cancer.scaled.train)  
cancer.scaled.lda.pred <- predict(cancer.scaled.lda, newdata = cancer.scaled.test)  
scaled.cm <- table(cancer.scaled.lda.pred$class, cancer.scaled.test.target)  
  
  
###################  
#unscaled data  
cancer.unscaled <- cancer\_data[,3:32]  
  
#split my data into training part and testing part  
cancer.unscaled.train <- cancer.unscaled[1:500,]  
cancer.unscaled.test <- cancer.unscaled[501:569,]  
cancer.unscaled.train.target <- cancer\_data[1:500,2]  
cancer.unscaled.test.target <- cancer\_data[501:569,2]  
  
#LDA using unscaling data  
cancer.unscaled.lda <- lda(cancer.unscaled.train.target~., data=cancer.unscaled.train)  
cancer.unscaled.lda.pred <- predict(cancer.unscaled.lda, newdata = cancer.unscaled.test)  
unscaled.cm <- table(cancer.unscaled.lda.pred$class, cancer.unscaled.test.target)  
  
##################  
#comparing models using differing type of data  
normalized.cm

## cancer.normalized.test.target  
## B M  
## B 39 4  
## M 0 26

scaled.cm

## cancer.scaled.test.target  
## B M  
## B 39 4  
## M 0 26

unscaled.cm

## cancer.unscaled.test.target  
## B M  
## B 39 4  
## M 0 26

scaling, normalization or leaving the data unscaled don't make a difference for LDA. They have the same result. The reason is that LDA decomposes ratio of Between-to-Within covariances and not the covariance itself having it magnitude.