HW5 - Jin Kweon - 3032235207

Jin Kweon

10/27/2017

The main purpose is to work around the concepts of how total dispersion can be broken down into between-groups and within-groups dispersion. These concepts are the root of linear discriminant analysis and quadratic discriminant analysis.

```
download.file("https://raw.githubusercontent.com/ucb-stat154/stat154-fall-2017/master/data/wine.data",
              destfile = "wine.txt")

wine <- read.table("wine.txt", header = T, sep = ",")
```

1) Sum-of-Squares Dispersion Functions (10 pts)

Remeber that TSS = BSS + WSS.

```
levels(as.factor(wine$class)) #class of each observation
```

```
## [1] "1" "2" "3"
```

```
#TSS function
tss <- function(x){
  ans <- sum((x - mean(x))^2)
  return(ans)
}

#Ans check
t <- tss(iris$Sepal.Length)
t
```

```
## [1] 102.1683
```

```
#BSS function
bss <- function(x, y){
  if(length(x) != length(y)){
    stop("predictor variable and response variable have different lengths...")
  }else{
    combined <- as.data.frame(cbind(y = y, x = x))
    combined$y <- as.factor(combined$y)
    splited <- split(combined, combined$y)
    xbar <- mean(combined$x)
    ans <- 0
```

```r
    for(i in 1:length(levels(combined$y))){
      ans <- ans + (nrow(splited[[i]]) * (mean(splited[[i]]$x) - xbar)^2)
    }
    return(ans)
  }
}

#Ans check
b <- bss(iris$Sepal.Length, iris$Species)
b
```

```
## [1] 63.21213
```

```r
#WSS function
wss <- function(x, y){
  if(length(x) != length(y)){
    stop("predictor variable and response variable have different lengths...")
  }else{
    combined <- as.data.frame(cbind(y = y, x = x))
    combined$y <- as.factor(combined$y)
    splited <- split(combined, combined$y)
    ans <- 0

    for(i in 1:length(levels(combined$y))){
      ans <- ans + sum((splited[[i]]$x - mean(splited[[i]]$x))^2)
    }
    return(ans)
  }
}

#Ans check
w <- wss(iris$Sepal.Length, iris$Species)
w
```

```
## [1] 38.9562
```

```r
#Check TSS = BSS + WSS
identical(b+w, t)
```

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

2) Sum-of-Squares Ratio Functions (10 pts)

```
#correlation ratio
cor_ratio <- function(x, y){
  b <- bss(x, y)
  t <- tss(x)
  cors <- b / t
  return(cors)
}

cor_ratio(iris$Sepal.Length, iris$Species)
```

```
## [1] 0.6187057
```

```
#F ratio
F_ratio <- function(x, y){
  combined <- as.data.frame(cbind(y = y, x = x))
  combined$y <- as.factor(combined$y)
  splited <- split(combined, combined$y)

  k <- length(levels(combined$y))
  n <- nrow(combined)

  b <- bss(x, y)
  w <- wss(x, y)
  f <- (b / (k - 1)) / (w / (n - k))
  return(f)
}

F_ratio(iris$Sepal.Length, iris$Species)
```

```
## [1] 119.2645
```

3) Discriminant Power of Predictors (30 pts)

Q. How to read the fit table here? ====> Algorithm to get optimization

Q. I dont understand what we are doing here… Whats the purpose of doing this ….??? ===> They are getting varaibles that discirimiate the classes the most

Q. what do you mean by the predictor is more discriminant? And, why the smaller the AIC, the more discriminant the predictor? Why the larger the F, the more discriminant the predictor? ====> Smaller AIC and larger F and R^2 ==> better disciminator. (AIC used the most!!!!) ==> AIC is telling me the better fit, and this process is telling me which variable is having better fit, and by doing that, we know which variable dicriminates the class better.

Q. Does multinom output $log\frac{p(y=k)}{p(y=K)}$? ===> glm and multinom outputs the coefficients.... So, this is equal to $log\frac{p}{1-p}$ for the simple logistic regression cuz p(y = K) is the # of variables = 2 for simple logistic regeression!!

Q. seems like correlation and AIC are more correlated than F and AIC, is it always true? ==> No it is not always the true...

Q. do we just rank it or do we have to change the order based on order? ==> it is better to change the order!!!

```
#Simple logistic regressions
x <- as.data.frame(wine[1:130,-1]) #get only class 1 and 2
aic <- data.frame(row.names = colnames(x), AIC = rep(0, ncol(x)))

for(i in 1:(length(colnames(wine))-1)){
  fit <- glm(as.factor(wine$class[1:130]) ~ x[1:130,i], family = binomial) #change to 1:130 if 1 and 2
  aic$AIC[i] <- fit$aic #same as extractAIC(fit)[2]
}

aic$Rank <- rank(aic$AIC)
aic
```

```
##                       AIC Rank
## alcohol          56.30075    2
## malic           182.85454   12
## ash             165.30370    9
## alcalinity      148.51462    6
## magnesium       162.10222    8
## phenols         139.62520    5
## flavanoids      121.51589    4
## nonflavanoids   166.94370   10
## proanthocyanins 174.71983   11
## color            81.96971    3
## hue             183.07125   13
## dilution        161.00793    7
## proline          45.21948    1
```

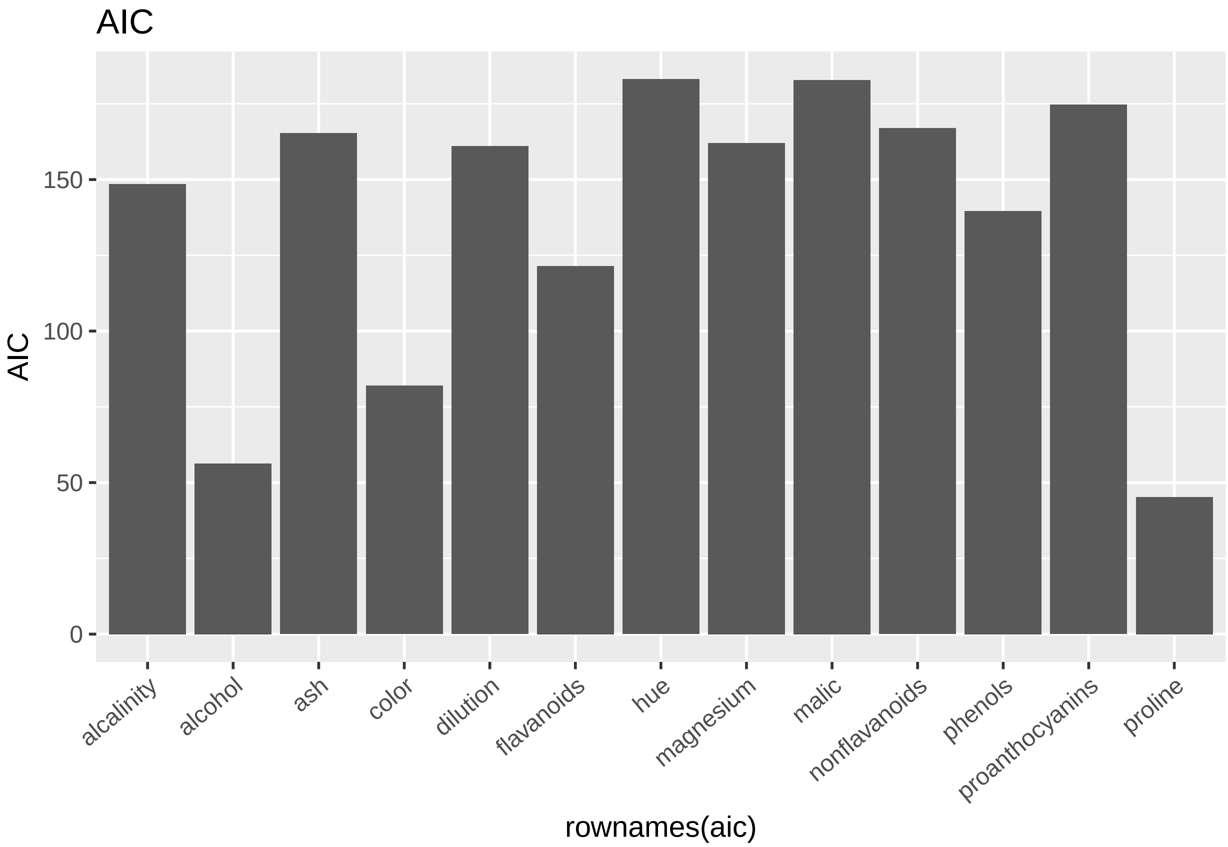
```
order <- aic[order(aic$Rank),] #They give the order number...
order
```

```
##                       AIC Rank
## proline          45.21948    1
## alcohol          56.30075    2
## color            81.96971    3
## flavanoids      121.51589    4
## phenols         139.62520    5
## alcalinity      148.51462    6
## dilution        161.00793    7
## magnesium       162.10222    8
## ash             165.30370    9
## nonflavanoids   166.94370   10
```

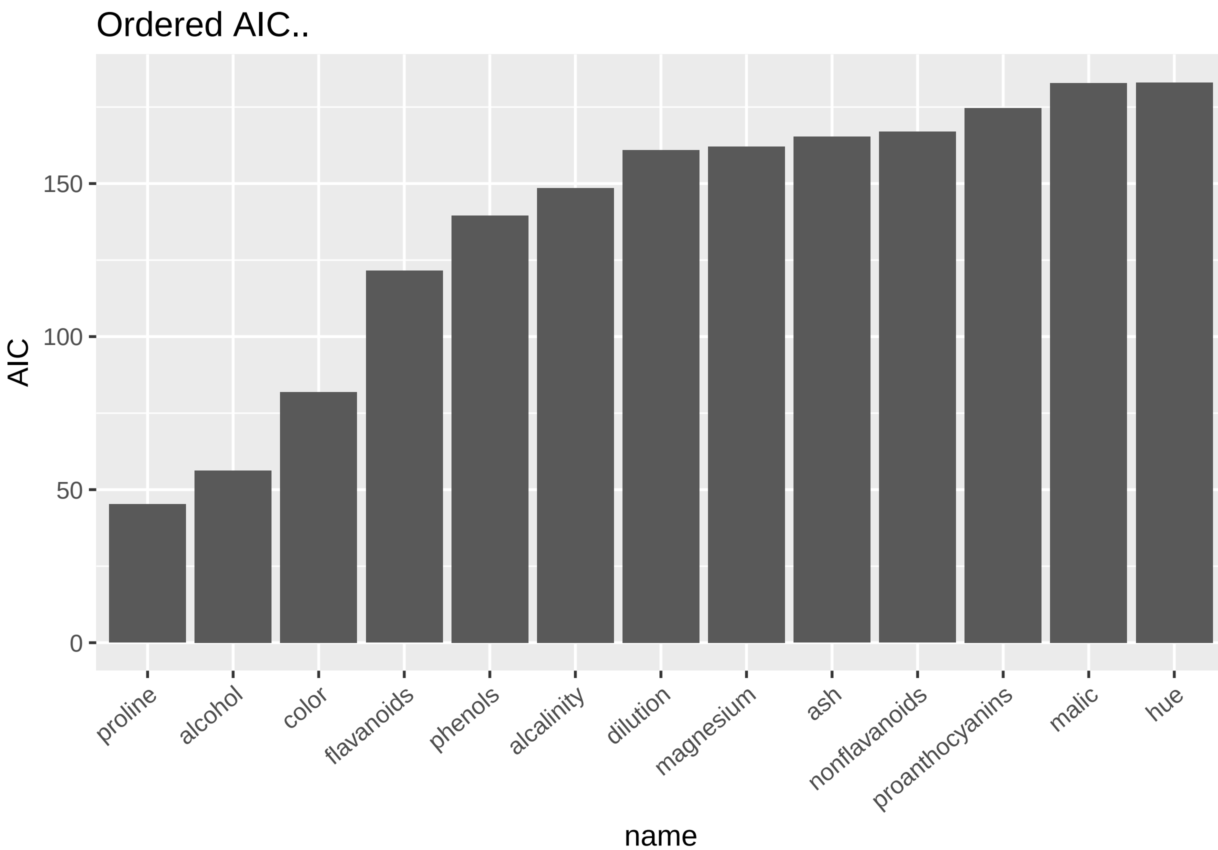
```
## proanthocyanins 174.71983   11
## malic           182.85454   12
## hue             183.07125   13
```

```r
order$name <- rownames(order)
order$name <- factor(order$name, levels = order$name[order(order$AIC)])

ggplot(aic, aes(x = rownames(aic), y = AIC)) + geom_bar(stat = "identity") + theme(axis.text.x = elemen
```




```r
ggplot(order, aes(x = name, y = AIC)) + geom_bar(stat = "identity") + theme(axis.text.x = element_text(
```




```
#Correlation ratios
correlation <- data.frame(row.names = colnames(x), cor = rep(0, ncol(x)))

for(i in 1:(length(colnames(wine))-1)){
  correlation$cor[i] <- cor_ratio(x[1:130,i], wine$class[1:130])
}

correlation$Rank <- rank(correlation$cor)
correlation
```

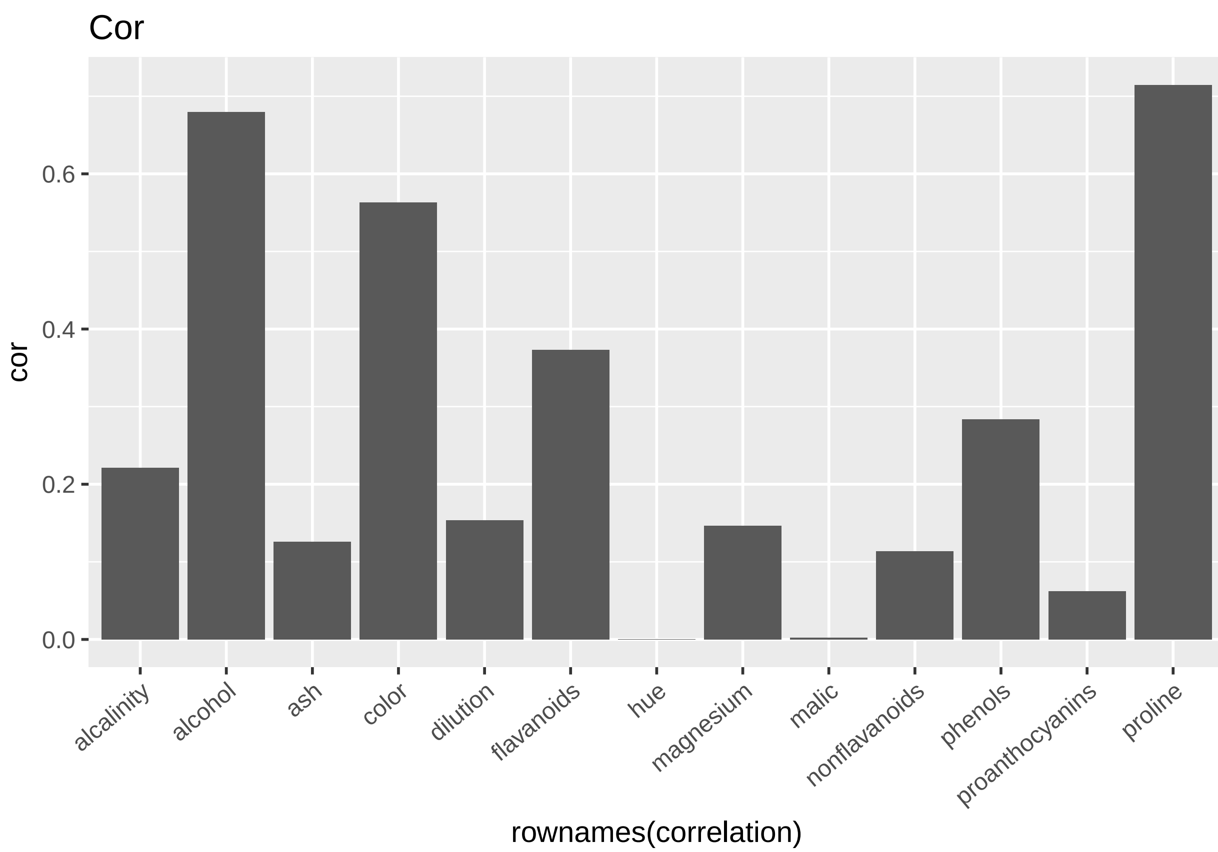
```
##                          cor Rank
## alcohol         0.6796337087   12
## malic           0.0019626987    2
## ash             0.1257044543    5
## alcalinity      0.2213110717    8
## magnesium       0.1467544634    6
## phenols         0.2837609465    9
## flavanoids      0.3729916215   10
## nonflavanoids   0.1138987546    4
## proanthocyanins 0.0621031600    3
## color           0.5634196215   11
## hue             0.0002904483    1
## dilution        0.1534649761    7
## proline         0.7145258216   13
```

```
order2 <- correlation[order(correlation$Rank),]
order2
```

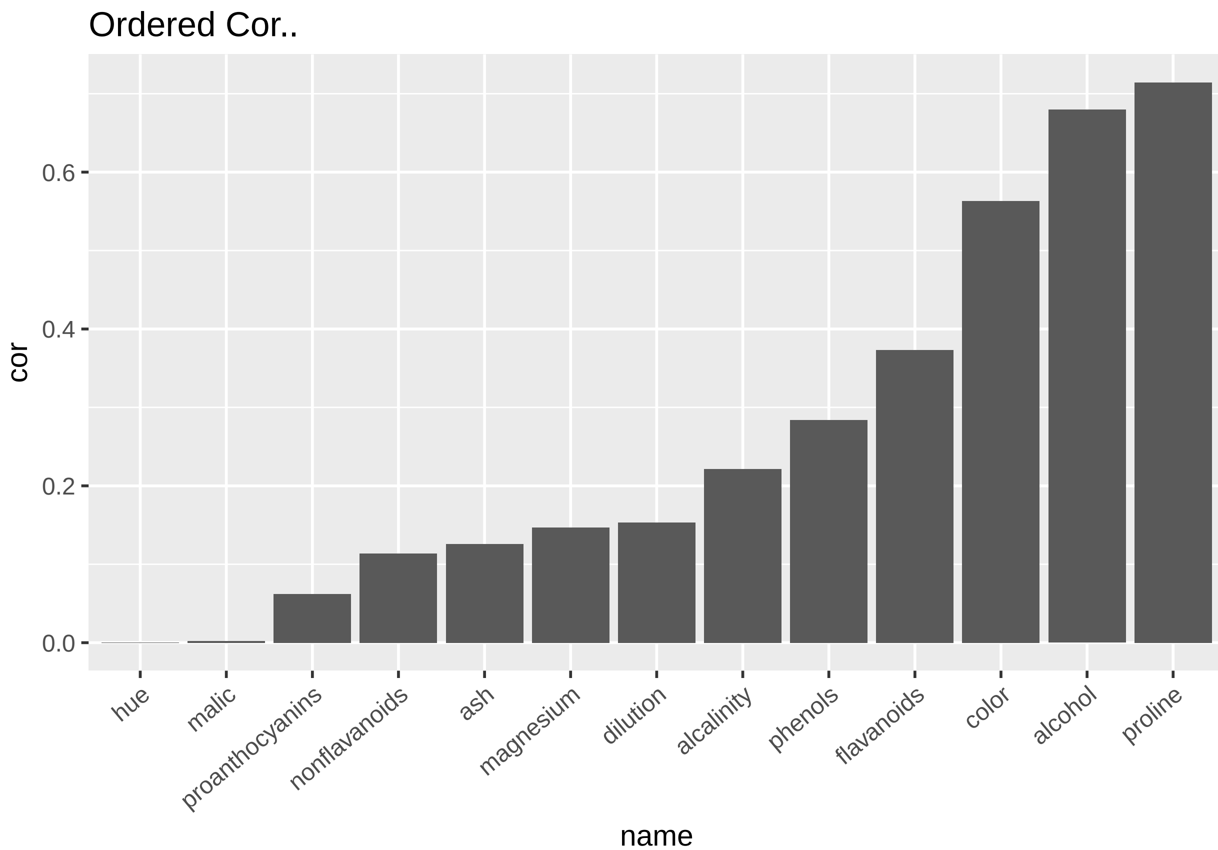
```
##                          cor Rank
## hue             0.0002904483    1
## malic           0.0019626987    2
## proanthocyanins 0.0621031600    3
## nonflavanoids   0.1138987546    4
## ash             0.1257044543    5
## magnesium       0.1467544634    6
## dilution        0.1534649761    7
## alcalinity      0.2213110717    8
## phenols         0.2837609465    9
## flavanoids      0.3729916215   10
## color           0.5634196215   11
## alcohol         0.6796337087   12
## proline         0.7145258216   13
```

```r
order2$name <- rownames(order2)
order2$name <- factor(order2$name, levels = order2$name[order(order2$cor)])

ggplot(correlation, aes(x = rownames(correlation), y = cor)) + geom_bar(stat = "identity") + theme(axis
```




```r
ggplot(order2, aes(x = name, y = cor)) + geom_bar(stat = "identity") + theme(axis.text.x = element_text
```




```
#F-ratios
Fratio <- data.frame(row.names = colnames(x), f = rep(0, ncol(x)))

for(i in 1:(length(colnames(wine))-1)){
  Fratio$f[i] <- F_ratio(x[1:130,i], wine$class[1:130])
}

Fratio$Rank <- rank(Fratio$f)
Fratio
```

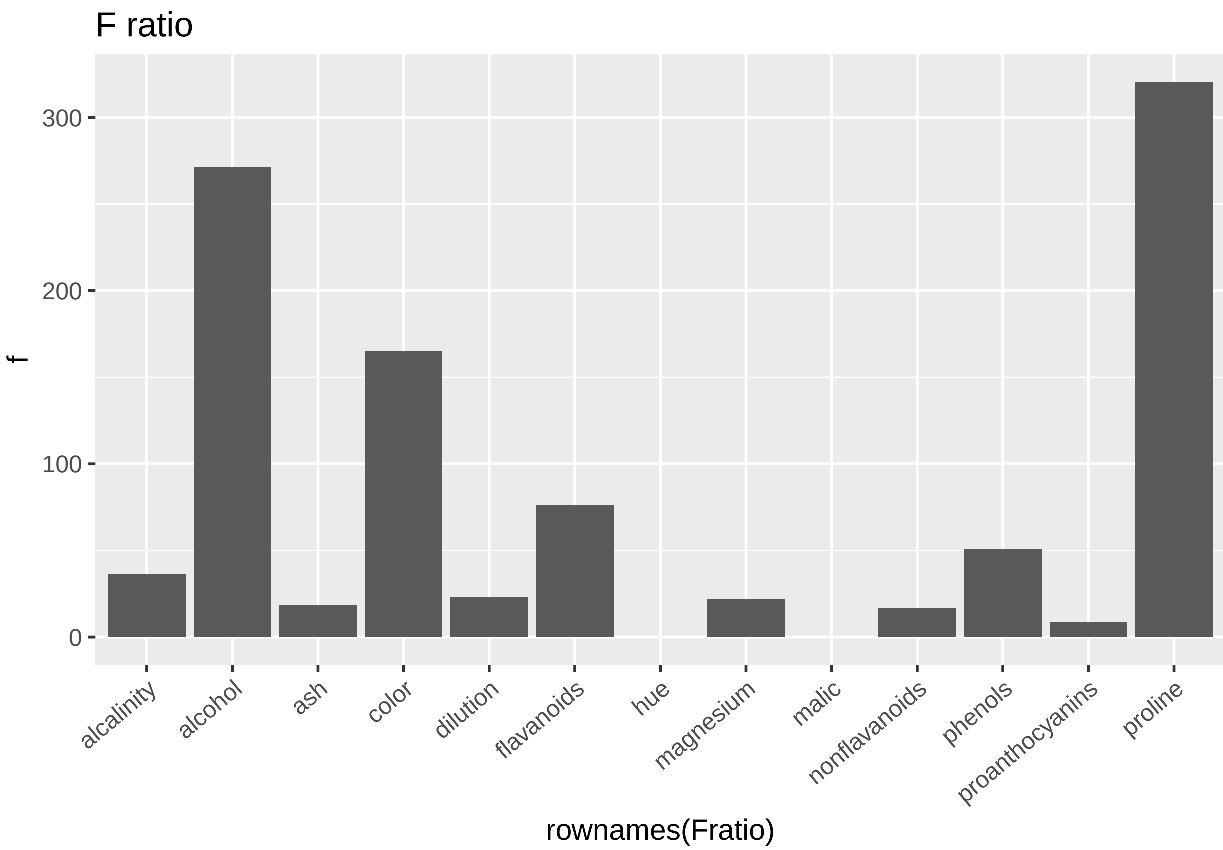
```
##                          f Rank
## alcohol         271.54265938   12
## malic             0.25171949    2
## ash              18.40358243    5
## alcalinity       36.37886215    8
## magnesium        22.01543461    6
## phenols          50.71128273    9
## flavanoids       76.14400251   10
## nonflavanoids    16.45301895    4
## proanthocyanins   8.47556377    3
## color           165.18770681   11
## hue               0.03718818    1
## dilution         23.20461220    7
## proline         320.37680494   13
```

```
order3 <- Fratio[order(Fratio$Rank), ]
order3
```

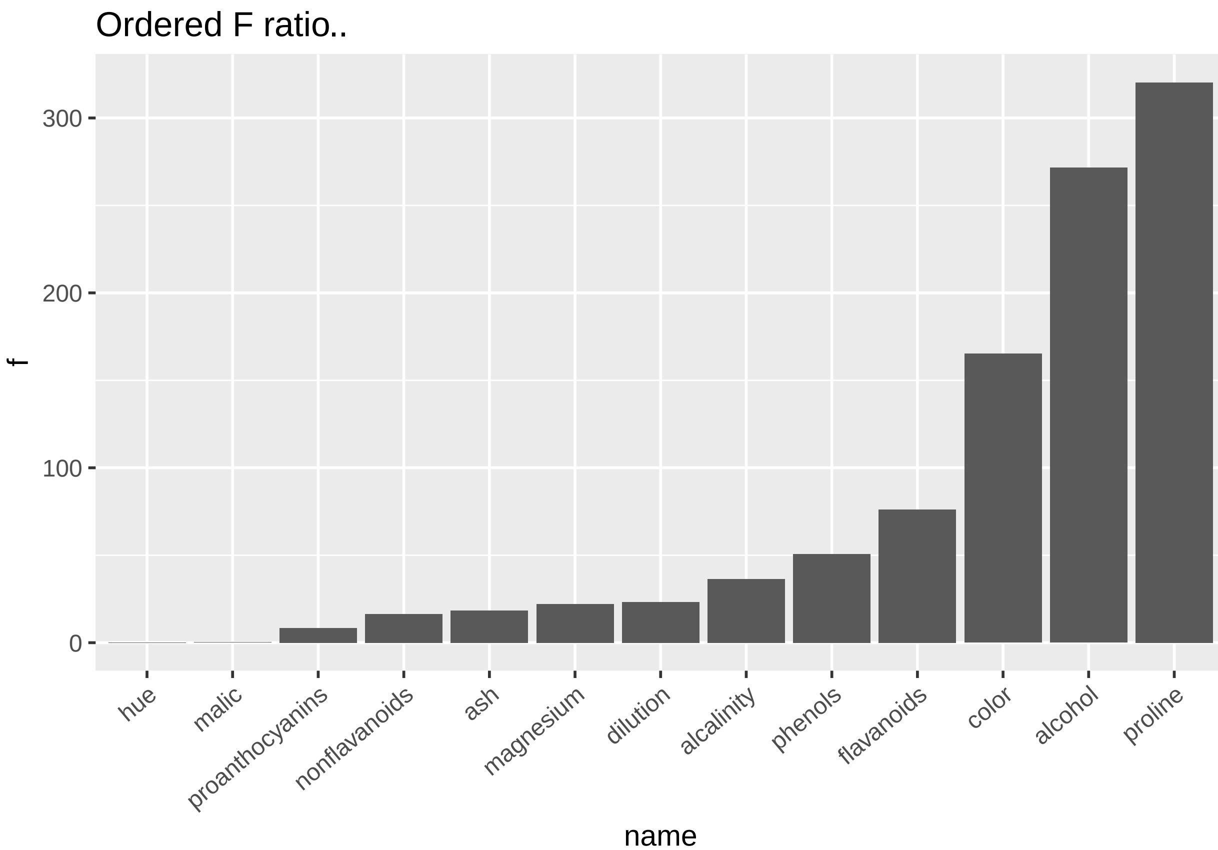
```
##                          f Rank
## hue              0.03718818    1
## malic            0.25171949    2
## proanthocyanins  8.47556377    3
## nonflavanoids   16.45301895    4
## ash             18.40358243    5
## magnesium       22.01543461    6
## dilution        23.20461220    7
## alcalinity      36.37886215    8
## phenols         50.71128273    9
## flavanoids      76.14400251   10
## color          165.18770681   11
## alcohol        271.54265938   12
## proline        320.37680494   13
```

```
order3$name <- rownames(order3)
order3$name <- factor(order3$name, levels = order3$name[order(order3$f)])

ggplot(Fratio, aes(x = rownames(Fratio), y = f)) + geom_bar(stat = "identity") + theme(axis.text.x = ele
```

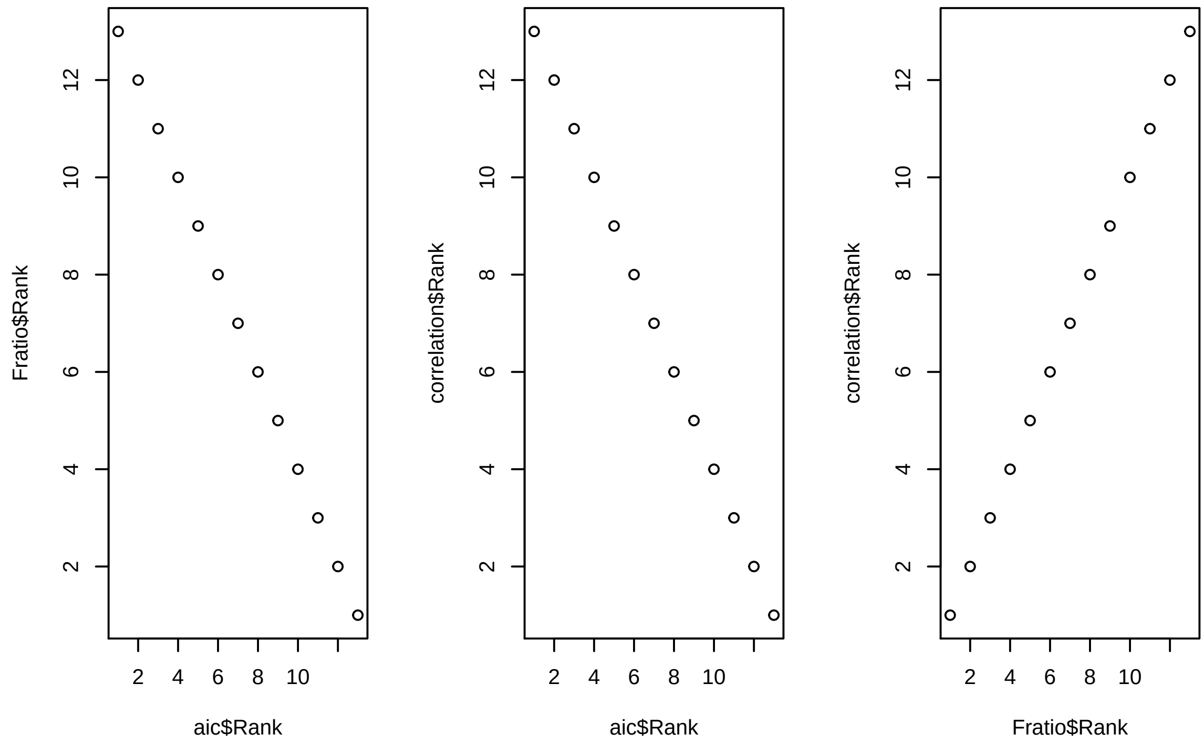



```
ggplot(order3, aes(x = name, y = f)) + geom_bar(stat = "identity") + theme(axis.text.x = element_text(an
```




```
# ggplot(Fratio, aes(x = rownames(Fratio), y = Rank)) + geom_bar(stat = "identity") + theme(axis.text.x


par(mfrow = c(1,3))
plot(aic$Rank, Fratio$Rank)
plot(aic$Rank, correlation$Rank)
plot(Fratio$Rank, correlation$Rank)
```



Comment:

For most of the times, the larger AIC, the smaller η^2 and F are. (meanign that η^2 and F are somewhat proportional)

Smaller AIC and larger F and R^2 ==> better disciminator (explnation above!!! - 3rd question!)

4) Variance functions (30 pts)

Total variance is $\frac{1}{n-1}X^TX$, where X is mean-centered.

Between variance is $\frac{1}{n-1}X^TY(Y^TY)^{-1}Y^TX$, where X is mean-centered.

Within variance is $\frac{1}{n-1}X^T(I-Y(Y^TY)^{-1}Y^T)X$

Q. so for BSS, BSS formula in pg 61 (slide 24) and pg 64 are the same. Right? ===> Right!!

```
#Total variance
total_variance <- function(predictors){
```

```r
  center <- scale(predictors, T, F)
  n <- nrow(predictors)
  v <- 1/(n-1) * t(center) %*% center
  return(v)
}

total_variance(iris[, 1:4])
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    0.6856935  -0.0424340    1.2743154   0.5162707
## Sepal.Width    -0.0424340   0.1899794   -0.3296564  -0.1216394
## Petal.Length    1.2743154  -0.3296564    3.1162779   1.2956094
## Petal.Width     0.5162707  -0.1216394    1.2956094   0.5810063
```

```r
#Check whether they are equal, elementwise
round(total_variance(iris[, 1:4]), 5) == round(var(iris[, 1:4]), 5)
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length         TRUE        TRUE         TRUE        TRUE
## Sepal.Width          TRUE        TRUE         TRUE        TRUE
## Petal.Length         TRUE        TRUE         TRUE        TRUE
## Petal.Width          TRUE        TRUE         TRUE        TRUE
```

```r
#Between variance
between_variance <- function(predictors, response){
  y <- dummy(response)
  x <- scale(predictors, T, F)
  x <- as.matrix(x)
  n <- nrow(predictors)
  b <- 1/(n-1) * t(x) %*% y %*% solve(t(y) %*% y) %*% t(y) %*% x
  return(b)
}

between_variance(iris[,1:4], iris$Species)
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    0.4242425 -0.13391051    1.1090497   0.4783848
## Sepal.Width    -0.1339105  0.07614049   -0.3841584  -0.1539105
## Petal.Length    1.1090497 -0.38415839    2.9335758   1.2535168
## Petal.Width     0.4783848 -0.15391051    1.2535168   0.5396868
```

```r
    #<<<< ..... Other way for Between variance ...... >>>>
    # x <- scale(iris[,1:4], T, F)
    # combined <- as.data.frame(cbind(y = y, x = x))
    # combined$y <- as.factor(combined$y)
    # splited <- split(combined, combined$y)
    #
    # g <- as.matrix(apply(x, 2, mean))
    #
    # ans <- matrix(0, 4, 4)
    # for(i in 1:length(levels(combined$y))){
    #   gk <- as.matrix(apply(splited[[i]][-1], 2, mean))
    #   ans <- ans + nrow(splited[[i]]) * tcrossprod(gk - g)
    # }
    # ans
    #
```

```r
#Within variance
within_variance <- function(predictors, response){
  y <- dummy(response)
  x <- scale(predictors, T, F)
  x <- as.matrix(x)
  n <- nrow(predictors)
  w <- 1/(n-1) * t(x) %*% (diag(n) - y %*% solve(t(y) %*% y) %*% t(y)) %*% x
  return(w)
}

within_variance(iris[,1:4], iris$Species)
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length   0.26145101  0.09147651   0.16526577  0.03788591
## Sepal.Width    0.09147651  0.11383893   0.05450201  0.03227114
## Petal.Length   0.16526577  0.05450201   0.18270201  0.04209262
## Petal.Width    0.03788591  0.03227114   0.04209262  0.04131946
```

```r
Viris <- total_variance(iris[,1:4])
Viris
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    0.6856935  -0.0424340    1.2743154   0.5162707
## Sepal.Width    -0.0424340   0.1899794   -0.3296564  -0.1216394
## Petal.Length    1.2743154  -0.3296564    3.1162779   1.2956094
## Petal.Width     0.5162707  -0.1216394    1.2956094   0.5810063
```

```r
Biris <- between_variance(iris[ ,1:4], iris$Species)
Wiris <- within_variance(iris[ ,1:4], iris$Species)
Biris + Wiris
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    0.6856935  -0.0424340    1.2743154   0.5162707
## Sepal.Width    -0.0424340   0.1899794   -0.3296564  -0.1216394
## Petal.Length    1.2743154  -0.3296564    3.1162779   1.2956094
## Petal.Width     0.5162707  -0.1216394    1.2956094   0.5810063
```

```r
#Check whether they are identical, elementwise
round(Viris, 5) == round(Biris + Wiris, 5)
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length         TRUE        TRUE         TRUE        TRUE
## Sepal.Width          TRUE        TRUE         TRUE        TRUE
## Petal.Length         TRUE        TRUE         TRUE        TRUE
## Petal.Width          TRUE        TRUE         TRUE        TRUE
```

5) Canonical Discriminant Analysis

- **Use the predictors and response of the wine data, to write code in R that allows you to find the eigenvectors uk. (20 pts)**

Steps:

1. Find C matrix (dimension of # of variables by # of groups)
2. Find W matrix & check B = CC^T
3. Get the e-vector of $C^T W^{-1} C$: w, through EVD.
4. Get the other matrix's e-vector: u.

Q. I did not really see the difference between symmetric = T or F inside of eigen function? What's the purpose of doing it, as x is already fixed to be symmetric or non-symmetric? ==> speed up the algorithm… & altho decimal points different they regard them as symmetric

Eigen value shows how valuable each column is…. If e-value is closed to 0, we do not need to use it….

Q. what is the point of asking for e-vector instead of e-values…??? ====> By getting w, we try to get u…. And, e-vector is telling us the direction, and e-value is telling us how valuable each component is….

```
head(wine)
```

```
##   class alcohol malic  ash alcalinity magnesium phenols flavanoids
## 1     1   14.23  1.71 2.43       15.6       127    2.80       3.06
## 2     1   13.20  1.78 2.14       11.2       100    2.65       2.76
## 3     1   13.16  2.36 2.67       18.6       101    2.80       3.24
## 4     1   14.37  1.95 2.50       16.8       113    3.85       3.49
## 5     1   13.24  2.59 2.87       21.0       118    2.80       2.69
## 6     1   14.20  1.76 2.45       15.2       112    3.27       3.39
##   nonflavanoids proanthocyanins color  hue dilution proline
## 1          0.28            2.29  5.64 1.04     3.92    1065
## 2          0.26            1.28  4.38 1.05     3.40    1050
## 3          0.30            2.81  5.68 1.03     3.17    1185
## 4          0.24            2.18  7.80 0.86     3.45    1480
## 5          0.39            1.82  4.32 1.04     2.93     735
## 6          0.34            1.97  6.75 1.05     2.85    1450
```

```
y <- wine[, 1]
x <- wine[, -1]


c <- matrix(0, dim(x)[2], length(levels(as.factor(y))))


getC <- function(x, y, c){
  combined <- as.data.frame(cbind(y = y, x = x))
  combined$y <- as.factor(combined$y)
  splited <- split(combined, combined$y)
```

```
  xjbar <- apply(x, 2, mean)
  n <- nrow(combined)

  for(i in 1:length(levels(combined$y))){
    coef <- nrow(splited[[i]]) / (n - 1)
    for(j in 1:dim(x)[2]){
      xkjbar <- mean(splited[[i]][,j + 1])
      c[j,i] <- sqrt(coef) * (xkjbar - xjbar[j])
    }
  }
  return(c)
}

C <- getC(x, y, c)
C
```

```
##               [,1]          [,2]        [,3]
##  [1,]   0.42962238 -4.572049e-01  0.07974436
##  [2,]  -0.18802586 -2.556651e-01  0.51940256
##  [3,]   0.05142826 -7.709629e-02  0.03674789
##  [4,]  -1.41892817  4.706311e-01  1.00074802
##  [5,]   3.80901645 -3.288519e+00 -0.22344220
##  [6,]   0.31468888 -2.295198e-02 -0.32097418
##  [7,]   0.55027440  3.266519e-02 -0.64980479
##  [8,]  -0.04148489  1.145118e-03  0.04460067
##  [9,]   0.17806819  2.494303e-02 -0.22775624
## [10,]   0.27147887 -1.248627e+00  1.21761007
## [11,]   0.06038187  6.259523e-02 -0.14307298
## [12,]   0.31529746  1.099915e-01 -0.48333608
## [13,] 212.93752144 -1.440147e+02 -60.92706944
```

```
dim(C)
```

```
## [1] 13  3
```

```
W <- within_variance(x, y)
B <- between_variance(x, y)
Total <- total_variance(y)

#Check B = Ct(C)
C %*% t(C)
```

```
##               [,1]         [,2]          [,3]        [,4]          [,5]
##  [1,]   0.39997090  0.077530654  0.0602739732 -0.74497416  3.122148e+00
##  [2,]   0.07753065  0.370497384  0.0291279353  0.66626232  8.509593e-03
##  [3,]   0.06027397  0.029127935  0.0099391114 -0.07248155  4.412127e-01
##  [4,]  -0.74497416  0.666262318 -0.0724815452  3.23634742 -7.176010e+00
##  [5,]   3.12214777  0.008509593  0.4412127133 -7.17600973  2.537289e+01
##  [6,]   0.12009526 -0.220016437  0.0061582909 -0.77853711  1.345852e+00
##  [7,]   0.16965724 -0.449327439  0.0019023356 -1.41571745  2.133778e+00
##  [8,]  -0.01478974  0.030673169 -0.0005827994  0.10403704 -1.717480e-01
##  [9,]   0.04693573 -0.158155660 -0.0011348389 -0.46885361  6.471294e-01
## [10,]   0.78460940  0.900615098  0.1549708025  0.24566906  4.868136e+00
## [11,]  -0.01408671 -0.101669240 -0.0069781557 -0.19939827  5.611844e-02
## [12,]   0.04662686 -0.338451063 -0.0100263180 -0.87931665  9.472617e-01
```

```
## [13,] 152.46834282 -34.863917059 19.8150605264 -430.89347054  1.298291e+03
##               [,6]          [,7]          [,8]         [,9]        [,10]
##  [1,]  0.120095263   0.169657238 -1.478974e-02  0.046935726   0.78460940
##  [2,] -0.220016437  -0.449327439  3.067317e-02 -0.158155660   0.90061510
##  [3,]  0.006158291   0.001902336 -5.827994e-04 -0.001134839   0.15497080
##  [4,] -0.778537109  -1.415717450  1.040370e-01 -0.468853608   0.24566906
##  [5,]  1.345852343   2.133777937 -1.717480e-01  0.647129378   4.86813647
##  [6,]  0.202580308   0.380986063 -2.739678e-02  0.128567459  -0.27673154
##  [7,]  0.380986063   0.726115198 -5.177240e-02  0.246798230  -0.68260763
##  [8,] -0.027396780  -0.051772397  3.711527e-03 -0.017516658   0.04161413
##  [9,]  0.128567459   0.246798230 -1.751666e-02  0.084203339  -0.26012108
## [10,] -0.276731541  -0.682607625  4.161413e-02 -0.260121081   3.11534464
## [11,]  0.063487550   0.128240789 -8.814407e-03  0.044899168  -0.23597280
## [12,]  0.251834484   0.491167117 -3.451124e-02  0.168970777  -0.64025665
## [13,] 89.870508136 152.060502347 -1.171599e+01 48.201757269 163.44321863
##              [,11]       [,12]       [,13]
##  [1,] -0.014086711  0.04662686   152.46834
##  [2,] -0.101669240 -0.33845106   -34.86392
##  [3,] -0.006978156 -0.01002632    19.81506
##  [4,] -0.199398269 -0.87931665  -430.89347
##  [5,]  0.056118435  0.94726168  1298.29118
##  [6,]  0.063487550  0.25183448    89.87051
##  [7,]  0.128240789  0.49116712   152.06050
##  [8,] -0.008814407 -0.03451124   -11.71599
##  [9,]  0.044899168  0.16897078    48.20176
## [10,] -0.235972799 -0.64025665   163.44322
## [11,]  0.028034010  0.09507553    12.55995
## [12,]  0.095075527  0.34512439    80.74652
## [13,] 12.559952329 80.74652230 69794.71551
```

```
B
```

```
##                      alcohol         malic           ash    alcalinity
## alcohol           0.39997090   0.077530654  0.0602739732   -0.74497416
## malic             0.07753065   0.370497384  0.0291279353    0.66626232
## ash               0.06027397   0.029127935  0.0099391114   -0.07248155
## alcalinity       -0.74497416   0.666262318 -0.0724815452    3.23634742
## magnesium         3.12214777   0.008509593  0.4412127133   -7.17600973
## phenols           0.12009526  -0.220016437  0.0061582909   -0.77853711
## flavanoids        0.16965724  -0.449327439  0.0019023356   -1.41571745
## nonflavanoids    -0.01478974   0.030673169 -0.0005827994    0.10403704
## proanthocyanins   0.04693573  -0.158155660 -0.0011348389   -0.46885361
## color             0.78460940   0.900615098  0.1549708025    0.24566906
## hue              -0.01408671  -0.101669240 -0.0069781557   -0.19939827
## dilution          0.04662686  -0.338451063 -0.0100263180   -0.87931665
## proline         152.46834282 -34.863917059 19.8150605264 -430.89347054
##                    magnesium      phenols    flavanoids nonflavanoids
## alcohol         3.122148e+00  0.120095263   0.169657238 -1.478974e-02
## malic           8.509593e-03 -0.220016437  -0.449327439  3.067317e-02
## ash             4.412127e-01  0.006158291   0.001902336 -5.827994e-04
## alcalinity     -7.176010e+00 -0.778537109  -1.415717450  1.040370e-01
## magnesium       2.537289e+01  1.345852343   2.133777937 -1.717480e-01
## phenols         1.345852e+00  0.202580308   0.380986063 -2.739678e-02
## flavanoids      2.133778e+00  0.380986063   0.726115198 -5.177240e-02
## nonflavanoids  -1.717480e-01 -0.027396780  -0.051772397  3.711527e-03
```

```
## proanthocyanins  6.471294e-01  0.128567459   0.246798230 -1.751666e-02
## color            4.868136e+00 -0.276731541  -0.682607625  4.161413e-02
## hue              5.611844e-02  0.063487550   0.128240789 -8.814407e-03
## dilution         9.472617e-01  0.251834484   0.491167117 -3.451124e-02
## proline          1.298291e+03 89.870508136 152.060502347 -1.171599e+01
##                 proanthocyanins        color          hue    dilution
## alcohol             0.046935726   0.78460940 -0.014086711  0.04662686
## malic              -0.158155660   0.90061510 -0.101669240 -0.33845106
## ash                -0.001134839   0.15497080 -0.006978156 -0.01002632
## alcalinity         -0.468853608   0.24566906 -0.199398269 -0.87931665
## magnesium           0.647129378   4.86813647  0.056118435  0.94726168
## phenols             0.128567459  -0.27673154  0.063487550  0.25183448
## flavanoids          0.246798230  -0.68260763  0.128240789  0.49116712
## nonflavanoids      -0.017516658   0.04161413 -0.008814407 -0.03451124
## proanthocyanins     0.084203339  -0.26012108  0.044899168  0.16897078
## color              -0.260121081   3.11534464 -0.235972799 -0.64025665
## hue                 0.044899168  -0.23597280  0.028034010  0.09507553
## dilution            0.168970777  -0.64025665  0.095075527  0.34512439
## proline            48.201757269 163.44321863 12.559952329 80.74652230
##                     proline
## alcohol           152.46834
## malic             -34.86392
## ash                19.81506
## alcalinity       -430.89347
## magnesium        1298.29118
## phenols            89.87051
## flavanoids        152.06050
## nonflavanoids     -11.71599
## proanthocyanins    48.20176
## color             163.44322
## hue                12.55995
## dilution           80.74652
## proline          69794.71551
```

```r
w <- eigen(t(C) %*% solve(W) %*% C)$vectors
eigen(t(C) %*% solve(W) %*% C)$values # the last e-value outputs 0....
```

```
## [1] 9.081739e+00 4.128469e+00 3.552714e-15
```

```r
u <- solve(W) %*% C %*% w
w
```

```
##             [,1]       [,2]       [,3]
## [1,]  0.65942356  0.4834252 -0.5757262
## [2,]  0.01685105 -0.7751385 -0.6315666
## [3,] -0.75158273  0.4067683 -0.5192908
```

```r
u
```

```
##                        [,1]          [,2]          [,3]
## alcohol           1.222609554  1.7814577940  4.857226e-16
## malic            -0.500847689  0.6240254979 -1.498801e-15
## ash               1.118580020  4.7936058021 -1.199041e-14
## alcalinity       -0.469155877 -0.2991204731  6.800116e-16
## magnesium         0.006557047 -0.0009456156  1.127570e-17
## phenols          -1.873169990 -0.0658249947  2.220446e-16
## flavanoids        5.034678679 -1.0053690476  3.996803e-15
```

```
## nonflavanoids    4.533472756 -3.3327580255  9.769963e-15
## proanthocyanins -0.406403118 -0.6275153789  1.561251e-15
## color           -1.076090082  0.5174619938 -1.776357e-15
## hue              2.479274325 -3.0971098347 -1.110223e-15
## dilution         3.508289345  0.1045914210 -2.220446e-15
## proline          0.008156412  0.0058299061 -4.553649e-18
```

- **Obtain the linear combinations zk and make a scatterplot of the wines. Add color to the dots indicating the different classes. (10 pts)**

Q. On the description, it says they try to separate group as much as possible, and the maximum number of linear combinations is k-1; however, we have k combinations here… Why?… And, what kind of linear combinations are they talking about…? ===> we do not use the third component…

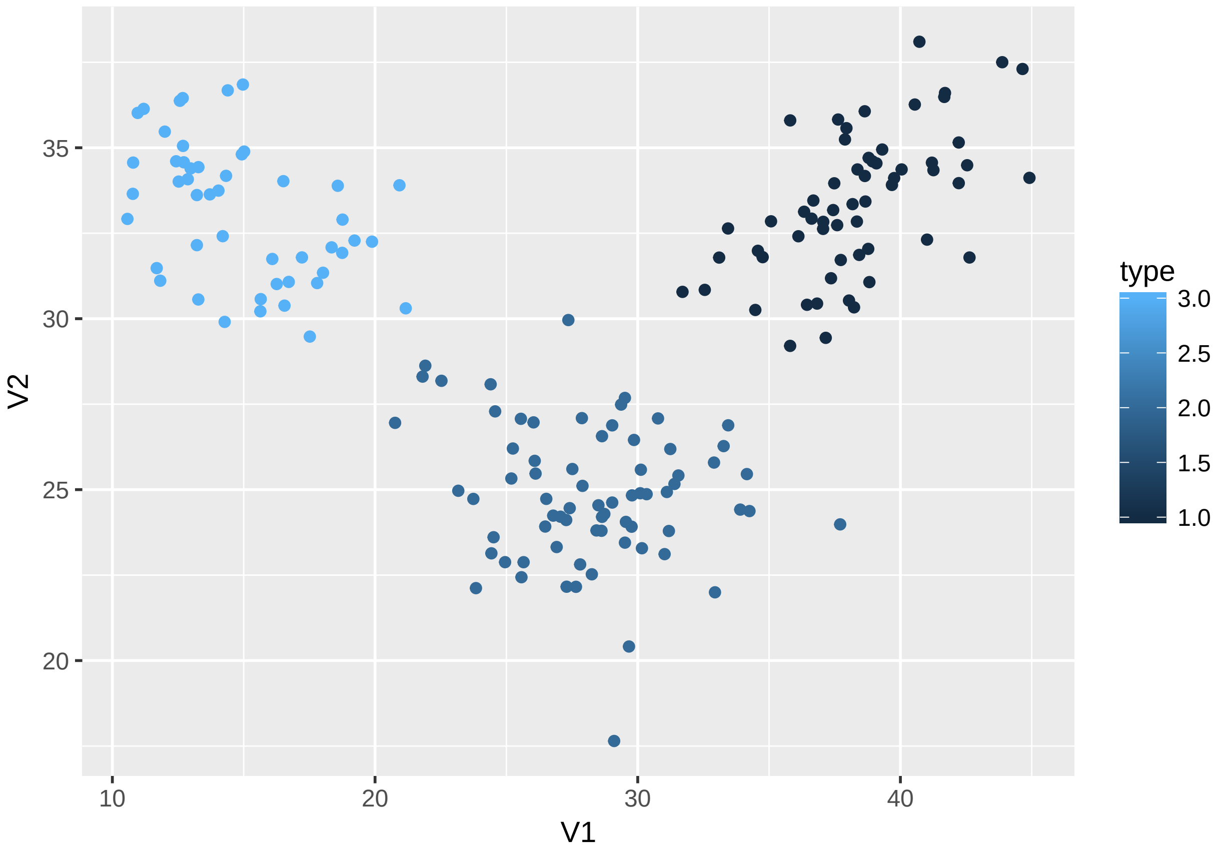
```r
z <- as.matrix(x) %*% u
z <- as.data.frame(z)

z$type <- wine$class
head(z, 10)
```

```
##          V1       V2            V3 type
## 1  42.22167 33.96474 -1.831189e-14    1
## 2  41.01456 32.31215 -1.829094e-14    1
## 3  38.34373 32.84077 -1.815104e-14    1
## 4  40.72298 38.10012 -2.187878e-14    1
## 5  32.55273 30.84253 -1.690907e-14    1
## 6  41.67142 36.48634 -1.893258e-14    1
## 7  41.69775 36.60074 -2.181146e-14    1
## 8  40.54900 36.26356 -2.275963e-14    1
## 9  39.67760 33.91211 -1.372622e-14    1
## 10 38.17978 33.35069 -1.882273e-14    1
```

```r
#splited2 <- split(z, z$type)

ggplot(data = z[,c(1, 2, 4)], aes(x = V1, y = V2, col = type)) + geom_point()
```

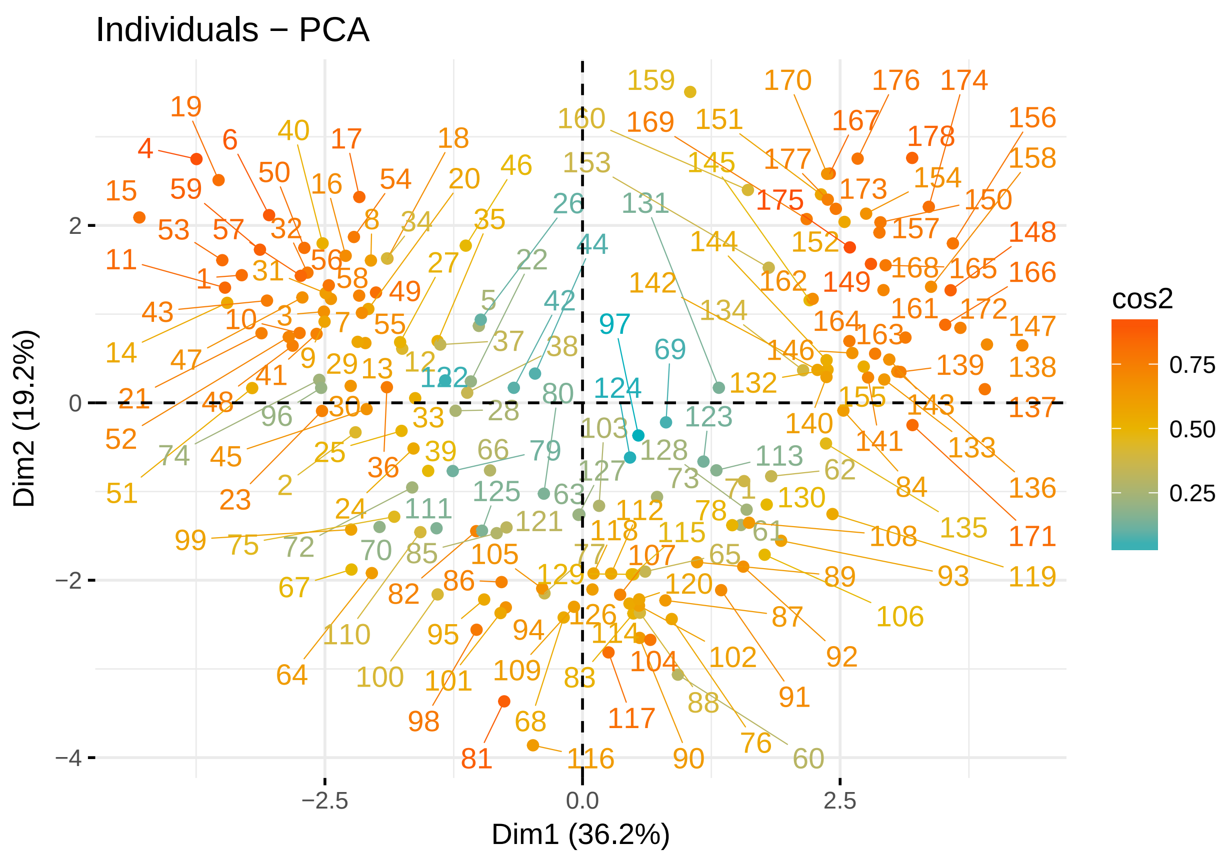



```
#ggplot(data = z[,c(1, 3, 4)], aes(x = V1, y = V3, col = type)) + geom_point() - last column has the e-

#ggplot(data = z[,c(2, 3, 4)], aes(x = V2, y = V3, col = type)) + geom_point() - last column has the e-
```

- **Obtain a scatterplot of the wines but this time using the first two principal components on the standardized predictors. Add color to the dots indicating the different classes. How does this compare to the previous scatterplot? (10 pts)**

```
winespca <- prcomp(wine[,-1], scale = TRUE)
fviz_pca_ind(winespca,
             col.ind = "cos2", # Color by the quality of representation
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```



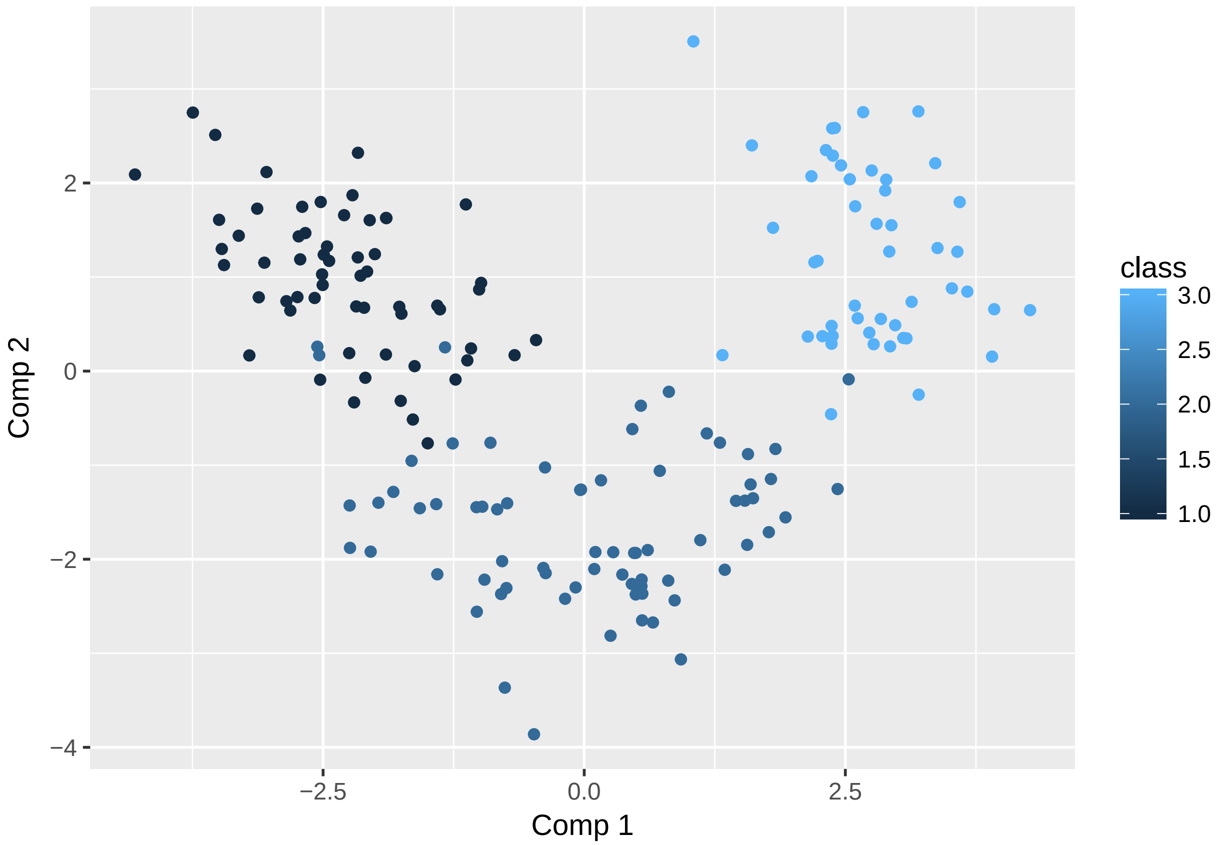

```r
winespca2 <- acp(wine[,-1])
winespca2$loadings
```

```
##                      Comp 1       Comp 2      Comp 3      Comp 4
## alcohol         -0.144329395  0.483651548  0.20738262 -0.01785630
## malic            0.245187580  0.224930935 -0.08901289  0.53689028
## ash              0.002051061  0.316068814 -0.62622390 -0.21417556
## alcalinity       0.239320405 -0.010590502 -0.61208035  0.06085941
## magnesium       -0.141992042  0.299634003 -0.13075693 -0.35179658
## phenols         -0.394660845  0.065039512 -0.14617896  0.19806835
## flavanoids      -0.422934297 -0.003359812 -0.15068190  0.15229479
## nonflavanoids    0.298533103  0.028779488 -0.17036816 -0.20330102
## proanthocyanins -0.313429488  0.039301722 -0.14945431  0.39905653
## color            0.088616705  0.529995672  0.13730621  0.06592568
## hue             -0.296714564 -0.279235148 -0.08522192 -0.42777141
## dilution        -0.376167411 -0.164496193 -0.16600459  0.18412074
## proline         -0.286752227  0.364902832  0.12674592 -0.23207086
##                     Comp 5      Comp 6      Comp 7      Comp 8
## alcohol         -0.26566365  0.21353865  0.05639636  0.39613926
## malic            0.03521363  0.53681385 -0.42052391  0.06582674
## ash             -0.14302547  0.15447466  0.14917061 -0.17026002
## alcalinity       0.06610294 -0.10082451  0.28696914  0.42797018
## magnesium        0.72704851  0.03814394 -0.32288330 -0.15636143
## phenols         -0.14931841 -0.08412230  0.02792498 -0.40593409
## flavanoids      -0.10902584 -0.01892002  0.06068521 -0.18724536
## nonflavanoids   -0.50070298 -0.25859401 -0.59544729 -0.23328465
## proanthocyanins  0.13685982 -0.53379539 -0.37213935  0.36822675
```

```
## color           -0.07643678 -0.41864414  0.22771214 -0.03379692
## hue             -0.17361452  0.10598274 -0.23207564  0.43662362
## dilution        -0.10116099  0.26585107  0.04476370 -0.07810789
## proline         -0.15786880  0.11972557 -0.07680450  0.12002267
##                      Comp 9     Comp 10     Comp 11     Comp 12
## alcohol         -0.50861912  0.21160473 -0.22591696 -0.26628645
## malic            0.07528304 -0.30907994  0.07648554  0.12169604
## ash              0.30769445 -0.02712539 -0.49869142 -0.04962237
## alcalinity      -0.20044931  0.05279942  0.47931378 -0.05574287
## magnesium       -0.27140257  0.06787022  0.07128891  0.06222011
## phenols         -0.28603452 -0.32013135  0.30434119 -0.30388245
## flavanoids      -0.04957849 -0.16315051 -0.02569409 -0.04289883
## nonflavanoids   -0.19550132  0.21553507  0.11689586  0.04235219
## proanthocyanins  0.20914487  0.13418390 -0.23736257 -0.09555303
## color           -0.05621752 -0.29077518  0.03183880  0.60422163
## hue             -0.08582839 -0.52239889 -0.04821201  0.25921400
## dilution        -0.13722690  0.52370587  0.04642330  0.60095872
## proline          0.57578611  0.16211600  0.53926983 -0.07940162
##                     Comp 13
## alcohol         -0.01496997
## malic           -0.02596375
## ash              0.14121803
## alcalinity      -0.09168285
## magnesium       -0.05677422
## phenols          0.46390791
## flavanoids      -0.83225706
## nonflavanoids   -0.11403985
## proanthocyanins  0.11691707
## color            0.01199280
## hue              0.08988884
## dilution         0.15671813
## proline         -0.01444734
```

```r
score <- as.data.frame(winespca2$scores)
score$class <- wine[,1]

ggplot(score, aes(x = `Comp 1`, y = `Comp 2`, col = class)) + geom_point()
```



Comment:

Since PCA does not have to do with y, and so PC is **not** a good way to separate the class. Although it looks similar with the previous one, this does not perfectly discriminate the groups…

- **Calculate the correlations between zk and the predictors. How do you interpret each score? (10 pts)**

Q. Can I say that a variable is more correlated with the LDA component where it has higher correlation??? ==> Right. And, we usually compare by row, not by column. So, we usually say that variable A is more correlated with the second component than variable B's.

Q. Do I make correlation with standardized predictors or the original? ==> cor funciton automatically standardize it…

```
class(z)
```

```
## [1] "data.frame"
```

```
class(x)
```

```
## [1] "data.frame"
```

```r
cor(z[,-c(3, 4)], x) #meaningless to get correlation for the last component..
```

```
##      alcohol      malic        ash alcalinity magnesium    phenols
## V1 0.2798969 -0.4891760 0.01918243 -0.5299978 0.1935927 0.75482118
## V2 0.8162180  0.3178155 0.40451247 -0.2148215 0.3355196 0.07008972
##     flavanoids nonflavanoids proanthocyanins      color        hue
## V1  0.89849357   -0.51522117      0.53203867 -0.3441133  0.6840759
## V2 -0.02635971   -0.02507846     -0.05042644  0.7665231 -0.3780354
##      dilution   proline
## V1  0.8503779 0.6148947
## V2 -0.2031988 0.6717132
```

Comment:

This correlation is showing how much each variable is correlated with each variable. So, the first component/score is somethinat that is correlated with phenols, flavanoids, and dilution. However, the second score is something that is correlated with alcohol, phenols, and color.

- **Create a matrix of size n × K, with the squared Mahalanobis distances d2(xi, gk) of each observation xi (i.e. each wine) to the each of the k centroids gk. The squared distance, with the Mahalanobis metric.**

Q. Centroid can be calculated as mean??? ==> YES!!

Q. So, the correlation from above question is to get how the variable is correlated with components, and this Mahalanobis distance is to get how each observation is related with each components/groups (as centroid is earned as we are averaging out all the observations from each component)? ==> Right!! Other way to see whether the observation i is belonged to which group, is to use Bayesian thing we did in the lab. . . .

```r
mah <- matrix(0, nrow(x), 3)
dim(mah)
```

```
## [1] 178   3
```

```r
W
```

```
##                       alcohol       malic          ash   alcalinity
## alcohol          0.2590914243  0.008080655 -0.013158814  -0.096118747
## malic            0.0080806554  0.877518019  0.021149104   0.410069396
## ash             -0.0131588142  0.021149104  0.065325524   0.478689823
## alcalinity      -0.0961187466  0.410069396  0.478689823   7.916338738
## magnesium        0.0177303467 -0.879289127  0.681723870   3.201249371
## phenols          0.0267919556 -0.014321286  0.015987300   0.107387963
## flavanoids       0.0223759839 -0.009302926  0.029632394   0.243634640
## nonflavanoids   -0.0009645222  0.010060193  0.006941271   0.046384817
## proanthocyanins  0.0165817948  0.017008679  0.002650417   0.091677388
## color            0.2436731397 -0.255776915  0.009683524  -0.100644873
## hue              0.0007732676 -0.041656398  0.002296001  -0.009719784
```

```
## dilution        -0.0049290331   0.046003580  0.010788154   0.223082279
## proline         12.0988421569 -32.684949506 -0.495321429 -32.461874477
##                    magnesium      phenols   flavanoids nonflavanoids
## alcohol           0.01773035  0.026791956  0.022375984 -0.0009645222
## malic            -0.87928913 -0.014321286 -0.009302926  0.0100601934
## ash               0.68172387  0.015987300  0.029632394  0.0069412708
## alcalinity        3.20124937  0.107387963  0.243634640  0.0463848174
## magnesium       178.61644239  0.570617536  0.659309094 -0.2838153497
## phenols           0.57061754  0.189109227  0.159484358 -0.0076483449
## flavanoids        0.65930909  0.159484358  0.271603475 -0.0150946029
## nonflavanoids    -0.28381535 -0.007648345 -0.015094603  0.0117771067
## proanthocyanins   1.28570310  0.090805886  0.126349323 -0.0085432104
## color             1.75238414  0.196734022  0.283438999 -0.0014936224
## hue               0.12473283 -0.001448674 -0.004158821  0.0013432303
## dilution         -0.27795361  0.059186795  0.067095138 -0.0099580031
## proline         470.86751731  8.300549123  3.386989876 -0.4875950903
##                 proanthocyanins        color           hue      dilution
## alcohol             0.016581795  0.243673140  0.0007732676  -0.004929033
## malic               0.017008679 -0.255776915 -0.0416563976   0.046003580
## ash                 0.002650417  0.009683524  0.0022960012   0.010788154
## alcalinity          0.091677388 -0.100644873 -0.0097197842   0.223082279
## magnesium           1.285703098  1.752384139  0.1247328312  -0.277953614
## phenols             0.090805886  0.196734022 -0.0014486740   0.059186795
## flavanoids          0.126349323  0.283438999 -0.0041588205   0.067095138
## nonflavanoids      -0.008543210 -0.001493622  0.0013432303  -0.009958003
## proanthocyanins     0.243391329  0.226617163 -0.0062346028   0.041962162
## color               0.226617163  2.259104741 -0.0405330022  -0.065555922
## hue                -0.006234603 -0.040533002  0.0242109504  -0.003309283
## dilution            0.041962162 -0.065555922 -0.0033092828   0.158962021
## proline            11.352576509 67.324261506  4.4402710570 -10.818996753
##                       proline
## alcohol            12.0988422
## malic             -32.6849495
## ash                -0.4953214
## alcalinity        -32.4618745
## magnesium         470.8675173
## phenols             8.3005491
## flavanoids          3.3869899
## nonflavanoids      -0.4875951
## proanthocyanins    11.3525765
## color              67.3242615
## hue                 4.4402711
## dilution          -10.8189968
## proline         29372.0018494
```

```r
splited3 <- split(wine, wine$class)


for(j in 1:3){
  gk <- apply(splited3[[j]][,-1], 2, mean)
  for(i in 1:nrow(x)){
    factor <- as.matrix(x[i, ] - gk)
    mah[i, j] <- factor %*% solve(W) %*% t(factor)
  }
```

```r
}
head(mah)
```

```
##           [,1]     [,2]     [,3]
## [1,] 11.471872 51.37512 92.28077
## [2,]  8.738074 39.13556 83.11946
## [3,]  7.884262 34.50203 68.51471
## [4,] 13.484011 67.09116 87.00835
## [5,] 11.668097 17.12809 42.12974
## [6,]  6.913637 55.98424 85.16075
```

- **Finally, assign each observation to the class Gk for which the Mahalanobis distance d2(xi, gk) is the smallest. And create a confussion matrix comparing the actual class versus the predicted class. (20 pts)**

```r
#Assigning
assign <- data.frame(observation = paste("observation",1:nrow(mah)), group = 0)
for(i in 1:nrow(mah)){
  assign[i,2] <- which.min(mah[i, ])
}
assign
```

```
##         observation group
## 1     observation 1     1
## 2     observation 2     1
## 3     observation 3     1
## 4     observation 4     1
## 5     observation 5     1
## 6     observation 6     1
## 7     observation 7     1
## 8     observation 8     1
## 9     observation 9     1
## 10   observation 10     1
## 11   observation 11     1
## 12   observation 12     1
## 13   observation 13     1
## 14   observation 14     1
## 15   observation 15     1
## 16   observation 16     1
## 17   observation 17     1
## 18   observation 18     1
## 19   observation 19     1
## 20   observation 20     1
## 21   observation 21     1
## 22   observation 22     1
```

```
## 23   observation 23      1
## 24   observation 24      1
## 25   observation 25      1
## 26   observation 26      1
## 27   observation 27      1
## 28   observation 28      1
## 29   observation 29      1
## 30   observation 30      1
## 31   observation 31      1
## 32   observation 32      1
## 33   observation 33      1
## 34   observation 34      1
## 35   observation 35      1
## 36   observation 36      1
## 37   observation 37      1
## 38   observation 38      1
## 39   observation 39      1
## 40   observation 40      1
## 41   observation 41      1
## 42   observation 42      1
## 43   observation 43      1
## 44   observation 44      1
## 45   observation 45      1
## 46   observation 46      1
## 47   observation 47      1
## 48   observation 48      1
## 49   observation 49      1
## 50   observation 50      1
## 51   observation 51      1
## 52   observation 52      1
## 53   observation 53      1
## 54   observation 54      1
## 55   observation 55      1
## 56   observation 56      1
## 57   observation 57      1
## 58   observation 58      1
## 59   observation 59      1
## 60   observation 60      2
## 61   observation 61      2
## 62   observation 62      2
## 63   observation 63      2
## 64   observation 64      2
## 65   observation 65      2
## 66   observation 66      2
## 67   observation 67      2
## 68   observation 68      2
## 69   observation 69      2
## 70   observation 70      2
## 71   observation 71      2
## 72   observation 72      2
## 73   observation 73      2
## 74   observation 74      2
## 75   observation 75      2
## 76   observation 76      2
```

```
## 77   observation 77      2
## 78   observation 78      2
## 79   observation 79      2
## 80   observation 80      2
## 81   observation 81      2
## 82   observation 82      2
## 83   observation 83      2
## 84   observation 84      2
## 85   observation 85      2
## 86   observation 86      2
## 87   observation 87      2
## 88   observation 88      2
## 89   observation 89      2
## 90   observation 90      2
## 91   observation 91      2
## 92   observation 92      2
## 93   observation 93      2
## 94   observation 94      2
## 95   observation 95      2
## 96   observation 96      2
## 97   observation 97      2
## 98   observation 98      2
## 99   observation 99      2
## 100 observation 100      2
## 101 observation 101      2
## 102 observation 102      2
## 103 observation 103      2
## 104 observation 104      2
## 105 observation 105      2
## 106 observation 106      2
## 107 observation 107      2
## 108 observation 108      2
## 109 observation 109      2
## 110 observation 110      2
## 111 observation 111      2
## 112 observation 112      2
## 113 observation 113      2
## 114 observation 114      2
## 115 observation 115      2
## 116 observation 116      2
## 117 observation 117      2
## 118 observation 118      2
## 119 observation 119      2
## 120 observation 120      2
## 121 observation 121      2
## 122 observation 122      2
## 123 observation 123      2
## 124 observation 124      2
## 125 observation 125      2
## 126 observation 126      2
## 127 observation 127      2
## 128 observation 128      2
## 129 observation 129      2
## 130 observation 130      2
```

```
## 131 observation 131     3
## 132 observation 132     3
## 133 observation 133     3
## 134 observation 134     3
## 135 observation 135     3
## 136 observation 136     3
## 137 observation 137     3
## 138 observation 138     3
## 139 observation 139     3
## 140 observation 140     3
## 141 observation 141     3
## 142 observation 142     3
## 143 observation 143     3
## 144 observation 144     3
## 145 observation 145     3
## 146 observation 146     3
## 147 observation 147     3
## 148 observation 148     3
## 149 observation 149     3
## 150 observation 150     3
## 151 observation 151     3
## 152 observation 152     3
## 153 observation 153     3
## 154 observation 154     3
## 155 observation 155     3
## 156 observation 156     3
## 157 observation 157     3
## 158 observation 158     3
## 159 observation 159     3
## 160 observation 160     3
## 161 observation 161     3
## 162 observation 162     3
## 163 observation 163     3
## 164 observation 164     3
## 165 observation 165     3
## 166 observation 166     3
## 167 observation 167     3
## 168 observation 168     3
## 169 observation 169     3
## 170 observation 170     3
## 171 observation 171     3
## 172 observation 172     3
## 173 observation 173     3
## 174 observation 174     3
## 175 observation 175     3
## 176 observation 176     3
## 177 observation 177     3
## 178 observation 178     3
```

```r
#Confusion matrix
assign[,2] == wine$class
```

```
##   [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [15] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [29] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
##  [43] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [57] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [71] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [85] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [99] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [113] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [127] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [141] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [155] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [169] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

```
assign$actual <- wine$class
assign
```

```
##         observation group actual
## 1     observation 1     1      1
## 2     observation 2     1      1
## 3     observation 3     1      1
## 4     observation 4     1      1
## 5     observation 5     1      1
## 6     observation 6     1      1
## 7     observation 7     1      1
## 8     observation 8     1      1
## 9     observation 9     1      1
## 10   observation 10     1      1
## 11   observation 11     1      1
## 12   observation 12     1      1
## 13   observation 13     1      1
## 14   observation 14     1      1
## 15   observation 15     1      1
## 16   observation 16     1      1
## 17   observation 17     1      1
## 18   observation 18     1      1
## 19   observation 19     1      1
## 20   observation 20     1      1
## 21   observation 21     1      1
## 22   observation 22     1      1
## 23   observation 23     1      1
## 24   observation 24     1      1
## 25   observation 25     1      1
## 26   observation 26     1      1
## 27   observation 27     1      1
## 28   observation 28     1      1
## 29   observation 29     1      1
## 30   observation 30     1      1
## 31   observation 31     1      1
## 32   observation 32     1      1
## 33   observation 33     1      1
## 34   observation 34     1      1
## 35   observation 35     1      1
## 36   observation 36     1      1
## 37   observation 37     1      1
## 38   observation 38     1      1
## 39   observation 39     1      1
## 40   observation 40     1      1
```

```
## 41   observation 41      1       1
## 42   observation 42      1       1
## 43   observation 43      1       1
## 44   observation 44      1       1
## 45   observation 45      1       1
## 46   observation 46      1       1
## 47   observation 47      1       1
## 48   observation 48      1       1
## 49   observation 49      1       1
## 50   observation 50      1       1
## 51   observation 51      1       1
## 52   observation 52      1       1
## 53   observation 53      1       1
## 54   observation 54      1       1
## 55   observation 55      1       1
## 56   observation 56      1       1
## 57   observation 57      1       1
## 58   observation 58      1       1
## 59   observation 59      1       1
## 60   observation 60      2       2
## 61   observation 61      2       2
## 62   observation 62      2       2
## 63   observation 63      2       2
## 64   observation 64      2       2
## 65   observation 65      2       2
## 66   observation 66      2       2
## 67   observation 67      2       2
## 68   observation 68      2       2
## 69   observation 69      2       2
## 70   observation 70      2       2
## 71   observation 71      2       2
## 72   observation 72      2       2
## 73   observation 73      2       2
## 74   observation 74      2       2
## 75   observation 75      2       2
## 76   observation 76      2       2
## 77   observation 77      2       2
## 78   observation 78      2       2
## 79   observation 79      2       2
## 80   observation 80      2       2
## 81   observation 81      2       2
## 82   observation 82      2       2
## 83   observation 83      2       2
## 84   observation 84      2       2
## 85   observation 85      2       2
## 86   observation 86      2       2
## 87   observation 87      2       2
## 88   observation 88      2       2
## 89   observation 89      2       2
## 90   observation 90      2       2
## 91   observation 91      2       2
## 92   observation 92      2       2
## 93   observation 93      2       2
## 94   observation 94      2       2
```

```
## 95   observation 95     2      2
## 96   observation 96     2      2
## 97   observation 97     2      2
## 98   observation 98     2      2
## 99   observation 99     2      2
## 100 observation 100     2      2
## 101 observation 101     2      2
## 102 observation 102     2      2
## 103 observation 103     2      2
## 104 observation 104     2      2
## 105 observation 105     2      2
## 106 observation 106     2      2
## 107 observation 107     2      2
## 108 observation 108     2      2
## 109 observation 109     2      2
## 110 observation 110     2      2
## 111 observation 111     2      2
## 112 observation 112     2      2
## 113 observation 113     2      2
## 114 observation 114     2      2
## 115 observation 115     2      2
## 116 observation 116     2      2
## 117 observation 117     2      2
## 118 observation 118     2      2
## 119 observation 119     2      2
## 120 observation 120     2      2
## 121 observation 121     2      2
## 122 observation 122     2      2
## 123 observation 123     2      2
## 124 observation 124     2      2
## 125 observation 125     2      2
## 126 observation 126     2      2
## 127 observation 127     2      2
## 128 observation 128     2      2
## 129 observation 129     2      2
## 130 observation 130     2      2
## 131 observation 131     3      3
## 132 observation 132     3      3
## 133 observation 133     3      3
## 134 observation 134     3      3
## 135 observation 135     3      3
## 136 observation 136     3      3
## 137 observation 137     3      3
## 138 observation 138     3      3
## 139 observation 139     3      3
## 140 observation 140     3      3
## 141 observation 141     3      3
## 142 observation 142     3      3
## 143 observation 143     3      3
## 144 observation 144     3      3
## 145 observation 145     3      3
## 146 observation 146     3      3
## 147 observation 147     3      3
## 148 observation 148     3      3
```

```
## 149 observation 149     3      3
## 150 observation 150     3      3
## 151 observation 151     3      3
## 152 observation 152     3      3
## 153 observation 153     3      3
## 154 observation 154     3      3
## 155 observation 155     3      3
## 156 observation 156     3      3
## 157 observation 157     3      3
## 158 observation 158     3      3
## 159 observation 159     3      3
## 160 observation 160     3      3
## 161 observation 161     3      3
## 162 observation 162     3      3
## 163 observation 163     3      3
## 164 observation 164     3      3
## 165 observation 165     3      3
## 166 observation 166     3      3
## 167 observation 167     3      3
## 168 observation 168     3      3
## 169 observation 169     3      3
## 170 observation 170     3      3
## 171 observation 171     3      3
## 172 observation 172     3      3
## 173 observation 173     3      3
## 174 observation 174     3      3
## 175 observation 175     3      3
## 176 observation 176     3      3
## 177 observation 177     3      3
## 178 observation 178     3      3
```

```
table(assign$group, assign$actual)
```

```
##
##      1  2  3
##   1 59  0  0
##   2  0 71  0
##   3  0  0 48
```