Lab 10: Canonical Discriminant Analysis

Donggyun Kim
27008257

4/2/2018

1) Sum-of-Squares Dispersion Functions

```
tss <- function(x) {
  sum((x - mean(x))^2)
}

tss(iris$Sepal.Length)
```

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

```
bss <- function(x, y) {
  if (length(x) != length(y)) {
    stop("length of x and y must be same!")
  }

  y <- factor(y)
  K <- levels(y)
  bss <- numeric(length(K))
  for (i in 1:length(K)) {
    index <- y == K[i]
    n <- sum(index)
    bss[i] <- n * (mean(x[index]) - mean(x))^2
  }
  sum(bss)
}

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

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

```
wss <- function(x, y) {
  if (length(x) != length(y)) {
    stop("length of x and y must be same!")
  }

  y <- factor(y)
  K <- levels(y)
  wss <- numeric(length(K))
  for (i in 1:length(K)) {
    index <- y == K[i]
    wss[i] <- sum((x[index] - mean(x[index]))^2)
  }

  sum(wss)
}
```

```r
wss(iris$Sepal.Length, iris$Species)
```

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

2) Sum-of-Squares Ratio Functions

```r
cor_ratio <- function(x, y) {
  bss(x, y) / tss(x)
}

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

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

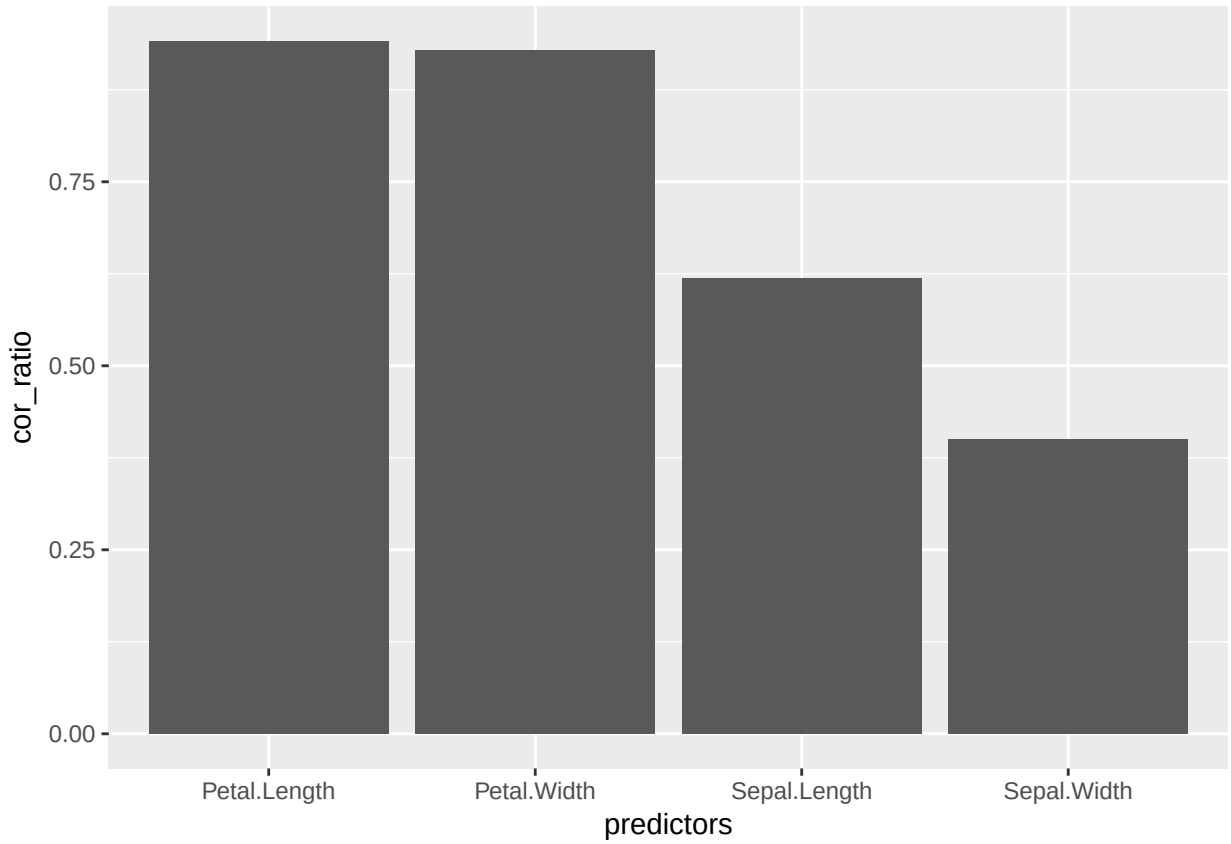
```r
F_ratio <- function(x, y) {
  y <- factor(y)
  K <- levels(y)

  bss(x, y) / wss(x, y) * (length(x) - length(K)) / (length(K) - 1)
}

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

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

3) Discriminant Power of Predictors

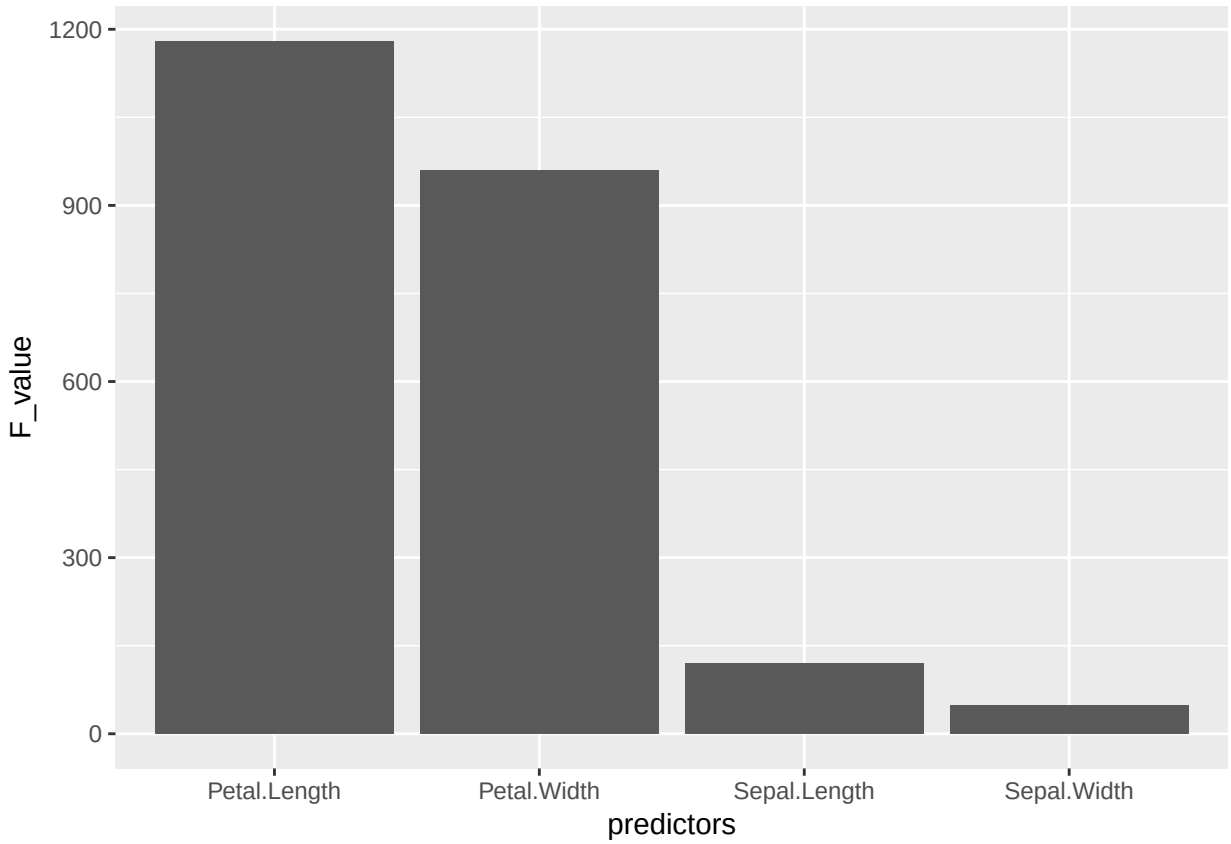
Correlation ratio

```r
y <- iris$Species
eta_square <- numeric(4)

for (i in 1:4) {
  x <- iris[, i]
  eta_square[i] <- cor_ratio(x, y)
}

dat <- data.frame(
  predictors = names(iris)[-5],
  cor_ratio = eta_square,
  rank = rank(eta_square)
)

library(ggplot2)
ggplot(dat, aes(x = predictors, y = cor_ratio)) +
  geom_bar(stat = "identity")
```



F-ratio

```
y <- iris$Species
F_value <- numeric(4)

for (i in 1:4) {
  x <- iris[, i]
  F_value[i] <- F_ratio(x, y)
}

dat <- data.frame(
  predictors = names(iris)[-5],
  cor_ratio = F_value,
  rank = rank(eta_square)
)

ggplot(dat, aes(x = predictors, y = F_value)) +
  geom_bar(stat = "identity")
```



4) Variance functions

```
total_variance <- function(X) {
  p <- dim(X)[2]
  n <- dim(X)[1]
  mat <- matrix(0, nrow = p, ncol = p)

  for (i in 1:p) {
    for (j in 1:p) {
      mat[i, j] <- t(X[, i] - mean(X[, i])) %*% (X[, j] - mean(X[, j]))
    }
  }

  mat / (n - 1)
}

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

```
##             [,1]       [,2]       [,3]       [,4]
## [1,]  0.6856935 -0.0424340  1.2743154  0.5162707
## [2,] -0.0424340  0.1899794 -0.3296564 -0.1216394
## [3,]  1.2743154 -0.3296564  3.1162779  1.2956094
## [4,]  0.5162707 -0.1216394  1.2956094  0.5810063
```

```r
var(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
between_variance <- function(X, y) {
  p <- dim(X)[2]
  n <- dim(X)[1]
  y <- factor(y)
  K <- levels(y)
  arr <- array(0, c(p, p, length(K)))

  for (k in 1:length(K)) {
    for (i in 1:p) {
      for (j in 1:p) {
        index <- y == K[k]
        arr[i, j, k] <- (mean(X[index, i]) - mean(X[, i])) *
          (mean(X[index, j]) - mean(X[, j])) * sum(index)
      }
    }
  }

  mat <- matrix(0, nrow = dim(arr)[1], ncol = dim(arr)[2])

  for (k in 1:dim(arr)[3]) {
    mat <- mat + arr[, , k]
  }

  mat / (n - 1)
}

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

```
##            [,1]        [,2]       [,3]       [,4]
## [1,]  0.4242425 -0.13391051  1.1090497  0.4783848
## [2,] -0.1339105  0.07614049 -0.3841584 -0.1539105
## [3,]  1.1090497 -0.38415839  2.9335758  1.2535168
## [4,]  0.4783848 -0.15391051  1.2535168  0.5396868
```

```r
within_variance <- function(X, y) {
  n <- dim(X)[1]
  p <- dim(X)[2]
  y <- factor(y)
  K <- levels(y)
  arr <- array(0, c(p, p, length(K)))

  for (k in 1:length(K)) {
    for (i in 1:p) {
      for (j in 1:p) {
        index <- y == K[k]
        arr[i, j, k] <- t(X[index, i] - mean(X[index, i])) %*% (X[index, j] - mean(X[index, j]))
      }
```

```
    }
  }

  mat <- matrix(0, nrow = p, ncol = p)

  for (k in 1:dim(arr)[3]) {
    mat <- mat + arr[, , k]
  }

  mat / (n - 1)
}

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

```
##             [,1]       [,2]       [,3]       [,4]
## [1,] 0.26145101 0.09147651 0.16526577 0.03788591
## [2,] 0.09147651 0.11383893 0.05450201 0.03227114
## [3,] 0.16526577 0.05450201 0.18270201 0.04209262
## [4,] 0.03788591 0.03227114 0.04209262 0.04131946
```

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

```
##             [,1]       [,2]       [,3]       [,4]
## [1,]  0.6856935 -0.0424340  1.2743154  0.5162707
## [2,] -0.0424340  0.1899794 -0.3296564 -0.1216394
## [3,]  1.2743154 -0.3296564  3.1162779  1.2956094
## [4,]  0.5162707 -0.1216394  1.2956094  0.5810063
```

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

```
##             [,1]       [,2]       [,3]       [,4]
## [1,]  0.6856935 -0.0424340  1.2743154  0.5162707
## [2,] -0.0424340  0.1899794 -0.3296564 -0.1216394
## [3,]  1.2743154 -0.3296564  3.1162779  1.2956094
## [4,]  0.5162707 -0.1216394  1.2956094  0.5810063
```

5) Canonical Discriminant Analysis (CDA)

```
p <- dim(iris)[2] - 1
n <- dim(iris)[1]
y <- iris$Species
K <- levels(y)
C <- matrix(0, nrow = p, ncol = length(K))

for (j in 1:p) {
  for (k in 1:length(K)) {
    index <- y == K[k]
    C[j, k] = sqrt(sum(index) / (n - 1)) * (mean(iris[index, j]) - mean(iris[, j]))
  }
}
```

```r
# compare to Biris
C %*% t(C)
```

```
##            [,1]        [,2]       [,3]       [,4]
## [1,]  0.4242425 -0.13391051  1.1090497  0.4783848
## [2,] -0.1339105  0.07614049 -0.3841584 -0.1539105
## [3,]  1.1090497 -0.38415839  2.9335758  1.2535168
## [4,]  0.4783848 -0.15391051  1.2535168  0.5396868
```

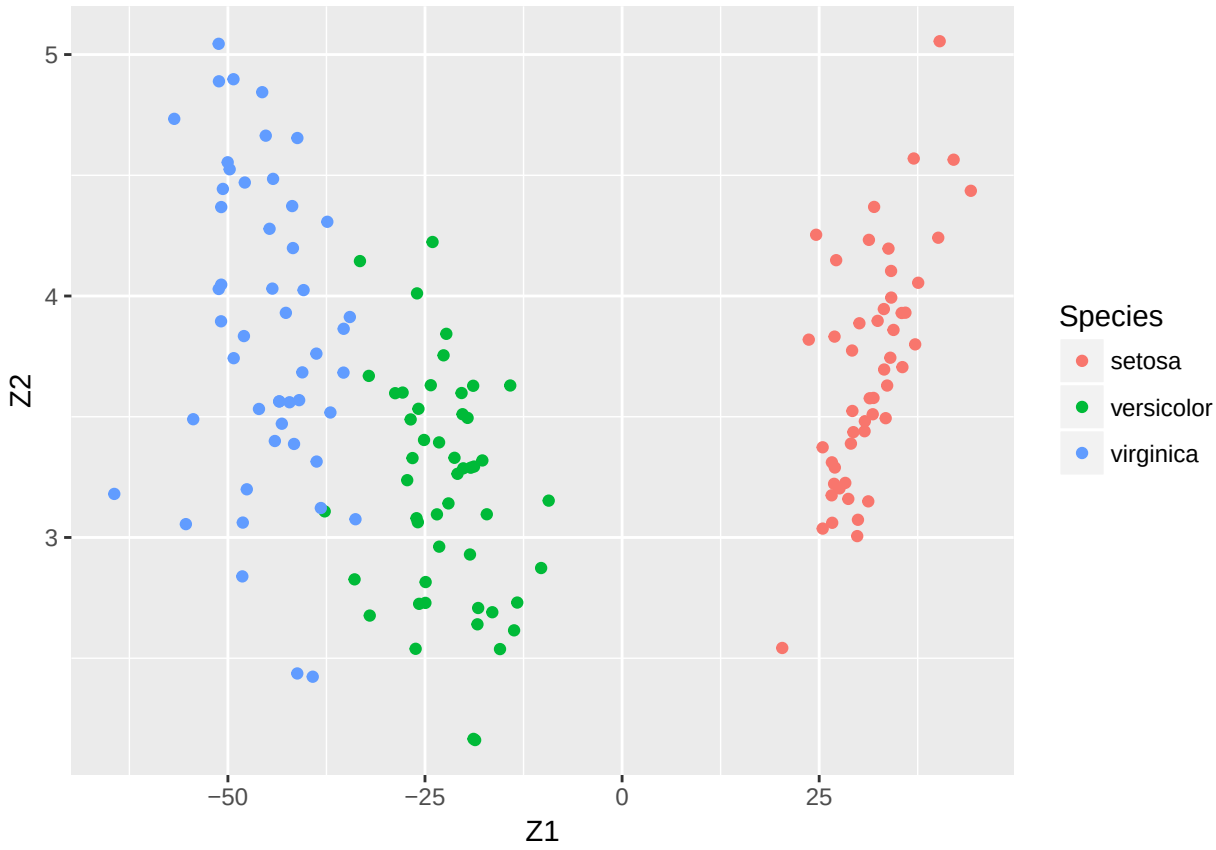
```r
Biris
```

```
##            [,1]        [,2]       [,3]       [,4]
## [1,]  0.4242425 -0.13391051  1.1090497  0.4783848
## [2,] -0.1339105  0.07614049 -0.3841584 -0.1539105
## [3,]  1.1090497 -0.38415839  2.9335758  1.2535168
## [4,]  0.4783848 -0.15391051  1.2535168  0.5396868
```

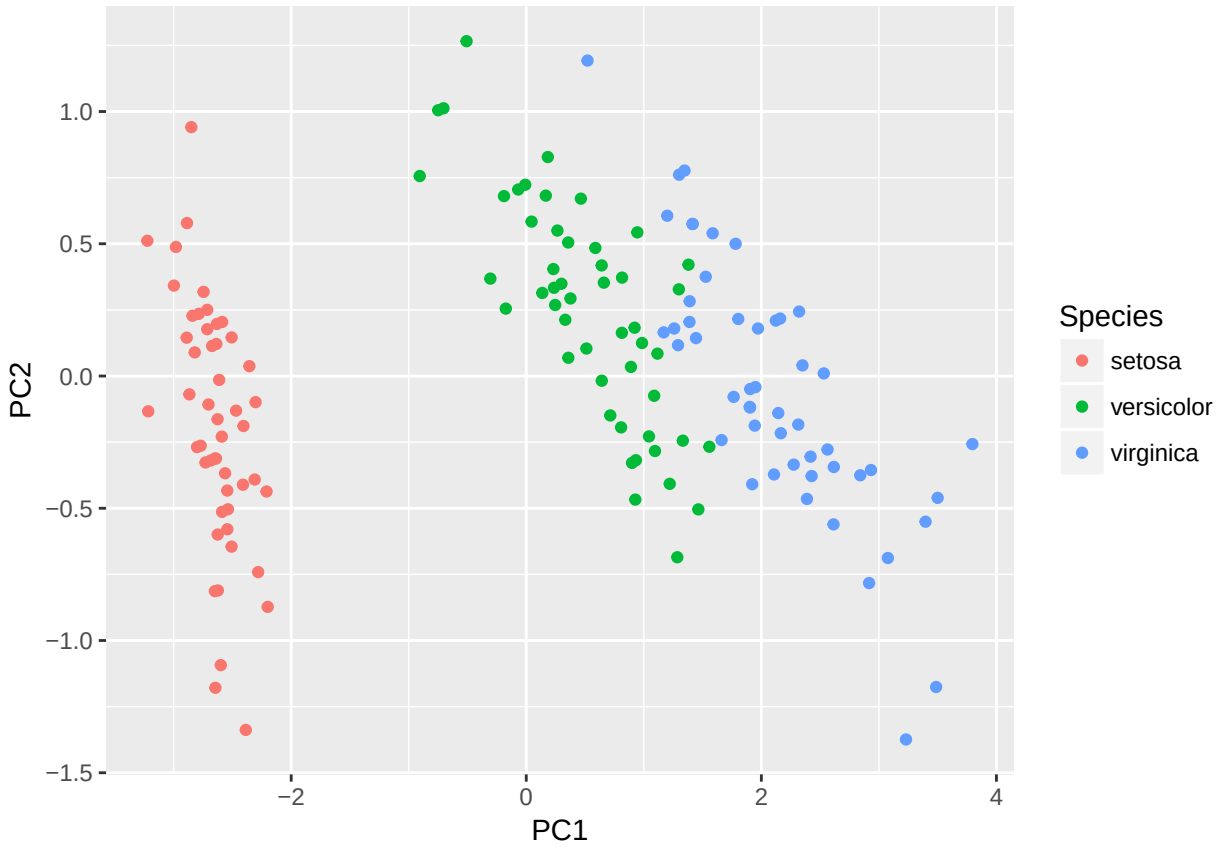
```r
EVD <- eigen(t(C) %*% solve(Wiris) %*% C)

w <- EVD$vectors

u <- solve(Wiris) %*% C %*% w
u
```

```
##            [,1]        [,2]          [,3]
## [1,]   4.737621  0.01296315  1.110223e-15
## [2,]   8.765309  1.16417022  8.881784e-15
## [3,] -12.573893 -0.50122628 -5.329071e-15
## [4,] -16.054080  1.52703420  1.865175e-14
```

```r
X <- as.matrix(iris[, 1:4])
Z <- X %*% u

dat <- data.frame(Z)
names(dat) <- paste0("Z", 1:3)
dat$Species <- iris$Species

ggplot(dat) +
  geom_point(aes(x = Z1, y = Z2, col = Species))
```




```
# PCA
pca <- prcomp(iris[, 1:4])
PCs <- pca$x

dat <- data.frame(PCs)
dat$Species <- iris$Species

ggplot(dat) +
  geom_point(aes(x = PC1, y = PC2, col = Species))
```


1.0
0.5
0.0
−0.5
−1.0
−1.5
PC2
−2
0
2
4
PC1
Species
setosa
versicolor
virginica