Assignment 8

Kaicheng Luo

2019/11/3

```
data <- iris
tss <- function(x){
  return(sum((x - mean(x))^2))
}
tss(iris$Sepal.Length)
```

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

```
bss <- function(x, group){
  if (length(x) != length(group)){
    stop()
  }
  group = factor(group)
  result = 0
  m <- mean(x)
  for (i in levels(group)){
    subgroup <- x[group == i]
    result = result + length(subgroup) * (mean(subgroup) - m)^2
  }
  return(result)
}
bss(iris$Sepal.Length, iris$Species)
```

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

```
wss <- function(x, group){
  if (length(x) != length(group)){
    stop()
  }
  group = factor(group)
  result = 0
  for (i in levels(group)){
    subgroup <- x[group == i]
    result = result + tss(subgroup)
  }
  return(result)
}
wss(iris$Sepal.Length, iris$Species)
```

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

```
# Note that by decomposition, this shall return zero
tss(iris$Sepal.Length) - wss(iris$Sepal.Length, iris$Species) - bss(iris$Sepal.Length, iris$Species)
```

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

```r
cor_ratio <- function(x, group){
  return(bss(x, group) / tss(x))
}
cor_ratio(iris$Sepal.Length, iris$Species)
```

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

```r
F_ratio <- function(x, group){
  return(
    bss(x, group) * (length(x) - length(levels(group))) / wss(x, group) * (length(levels(group)) - 1)
  )
}
F_ratio(iris$Sepal.Length, iris$Species)
```

```
## [1] 477.058
```

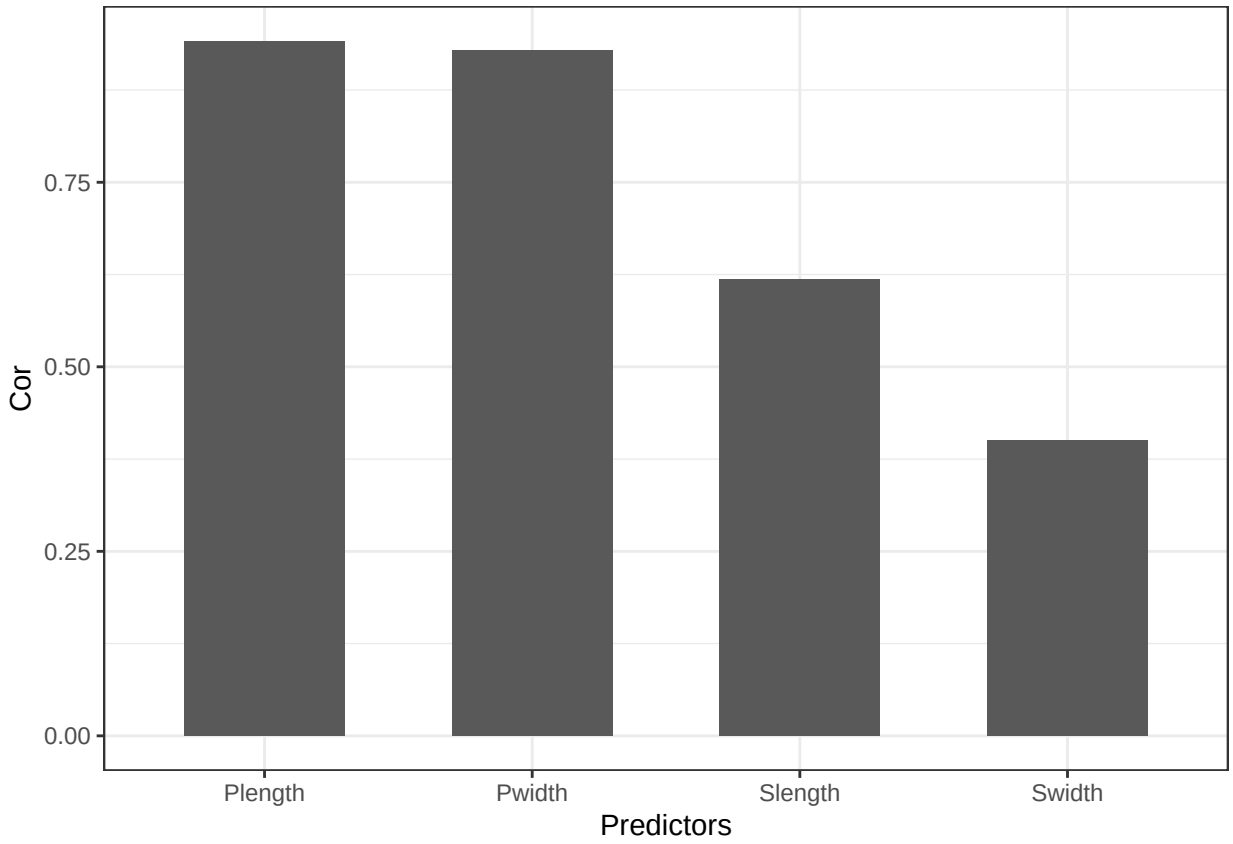
```r
list("Slength" = cor_ratio(iris$Sepal.Length, iris$Species), "Swidth" = cor_ratio(iris$Sepal.Width, iris
```

```
## $Slength
## [1] 0.6187057
##
## $Swidth
## [1] 0.4007828
##
## $Plength
## [1] 0.9413717
##
## $Pwidth
## [1] 0.9288829
```

```r
data.frame("Predictors" = c("Slength","Swidth","Plength","Pwidth"), "Cor" = c(cor_ratio(iris$Sepal.Leng
```

```
##   Predictors       Cor
## 1     Swidth 0.4007828
## 2    Slength 0.6187057
## 3     Pwidth 0.9288829
## 4    Plength 0.9413717
```

```r
data.frame("Predictors" = c("Slength","Swidth","Plength","Pwidth"), "Cor" = c(cor_ratio(iris$Sepal.Leng
  ggplot() + theme_bw() +
    geom_bar(aes(x = Predictors, y = Cor), stat = "identity", width = .6)
```



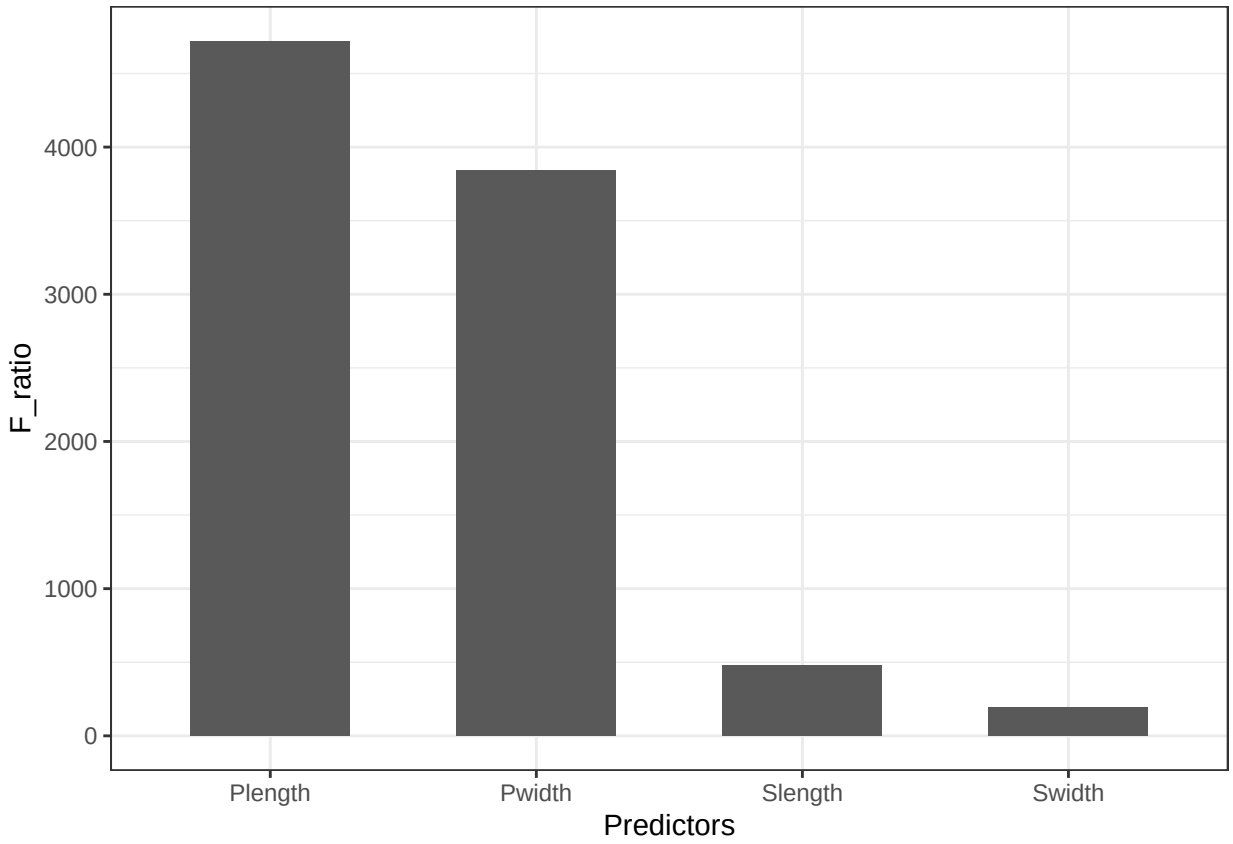

```
list("Slength" = F_ratio(iris$Sepal.Length, iris$Species), "Swidth" = F_ratio(iris$Sepal.Width, iris$Sp
```

```
## $Slength
## [1] 477.058
##
## $Swidth
## [1] 196.6402
##
## $Plength
## [1] 4720.645
##
## $Pwidth
## [1] 3840.029
```

```
data.frame("Predictors" = c("Slength","Swidth","Plength","Pwidth"), "F_ratio" = c(F_ratio(iris$Sepal.Le
```

```
##   Predictors   F_ratio
## 1     Swidth  196.6402
## 2    Slength  477.0580
## 3     Pwidth 3840.0286
## 4    Plength 4720.6447
```

```
data.frame("Predictors" = c("Slength","Swidth","Plength","Pwidth"), "F_ratio" = c(F_ratio(iris$Sepal.Le
  ggplot() + theme_bw() +
    geom_bar(aes(x = Predictors, y = F_ratio), stat = "identity", width = .6)
```




```
total_variance <- function(x){
  X <- as.matrix(x - (matrix(1, nrow = nrow(x)) %*% t(apply(x, 2, mean))))
  return(t(X) %*% X / (nrow(X)-1))
}
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
```

```
# Compare with the built-in functions, they are identical!
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
```

```
between_variance <- function(x, group){
  m <- apply(x, 2, mean)
  group <- factor(group)
  result = matrix(0, nrow = ncol(x), ncol = ncol(x))
```

```
  for (i in levels(group)){
    subgroup = x[group == i, ]
    result = result + nrow(subgroup) / (nrow(x)-1) *(apply(subgroup, 2, mean) - m) %*% t(apply(subgroup
  }
  return(result)
}
between_variance(iris[,1:4], iris[,5])
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width
## [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
```

```
within_variance <- function(x, group){
  group <- factor(group)
  result = matrix(0, nrow = ncol(x), ncol = ncol(x))
  for (i in levels(group)){
    subgroup = x[group == i,]
    result = result + (nrow(subgroup)-1) * total_variance(subgroup) / (nrow(x) - 1)
  }
  return(result)
}
within_variance(iris[ ,1:4], iris[ ,5])
```

```
##              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
```

```
# Here we can verify our decompostion formula
round(total_variance(iris[,1:4]) - within_variance(iris[ ,1:4], iris[ ,5]) - between_variance(iris[,1:4
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length            0           0            0           0
## Sepal.Width             0           0            0           0
## Petal.Length            0           0            0           0
## Petal.Width             0           0            0           0
```

```
# confirm V = B + W
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
#B+W
Biris <- between_variance(iris[ ,1:4], iris$Species)
Wiris <- within_variance(iris[ ,1:4], iris$Species)
Biris + Wiris
```

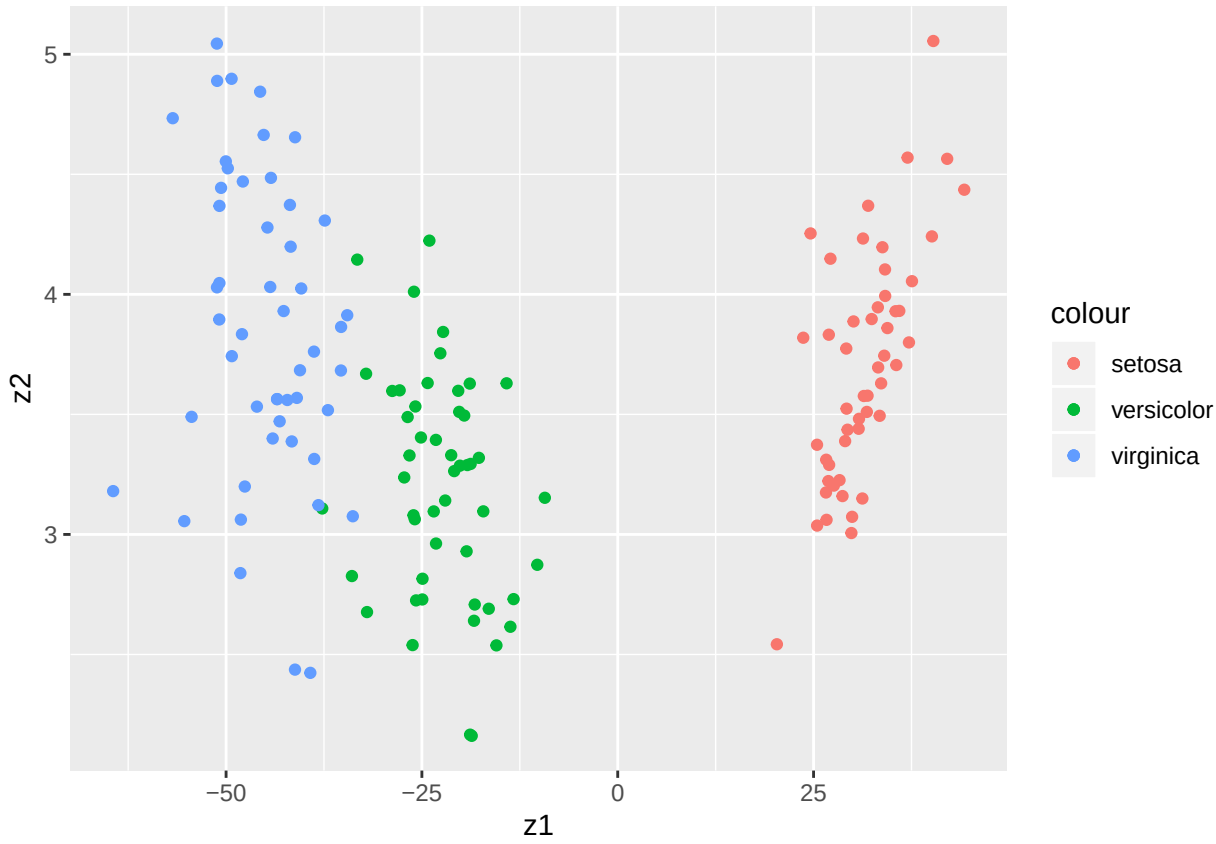
```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width
## [1,]    0.6856935  -0.0424340    1.2743154   0.5162707
## [2,]   -0.0424340   0.1899794   -0.3296564  -0.1216394
## [3,]    1.2743154  -0.3296564    3.1162779   1.2956094
## [4,]    0.5162707  -0.1216394    1.2956094   0.5810063
```

```r
C <- function(x, group){
  m <- apply(x, 2, mean)
  group <- factor(group)
  result = matrix(0, nrow = ncol(x), ncol = ncol(x))
  for (j in 1:ncol(x)){
    k = 0
    for (i in levels(group)){
      subgroup = x[group == i, ]
      k = k+1
      result[j,k] = sqrt(nrow(subgroup) / (nrow(x)-1)) * (mean(subgroup[,j]) - m[j])
    }
  }
  return(result)
}
# Verify that the decompostion is correct
round(C(iris[ ,1:4], iris$Species) %*% t(C(iris[ ,1:4], iris$Species)) -between_variance(iris[ ,1:4], i
```

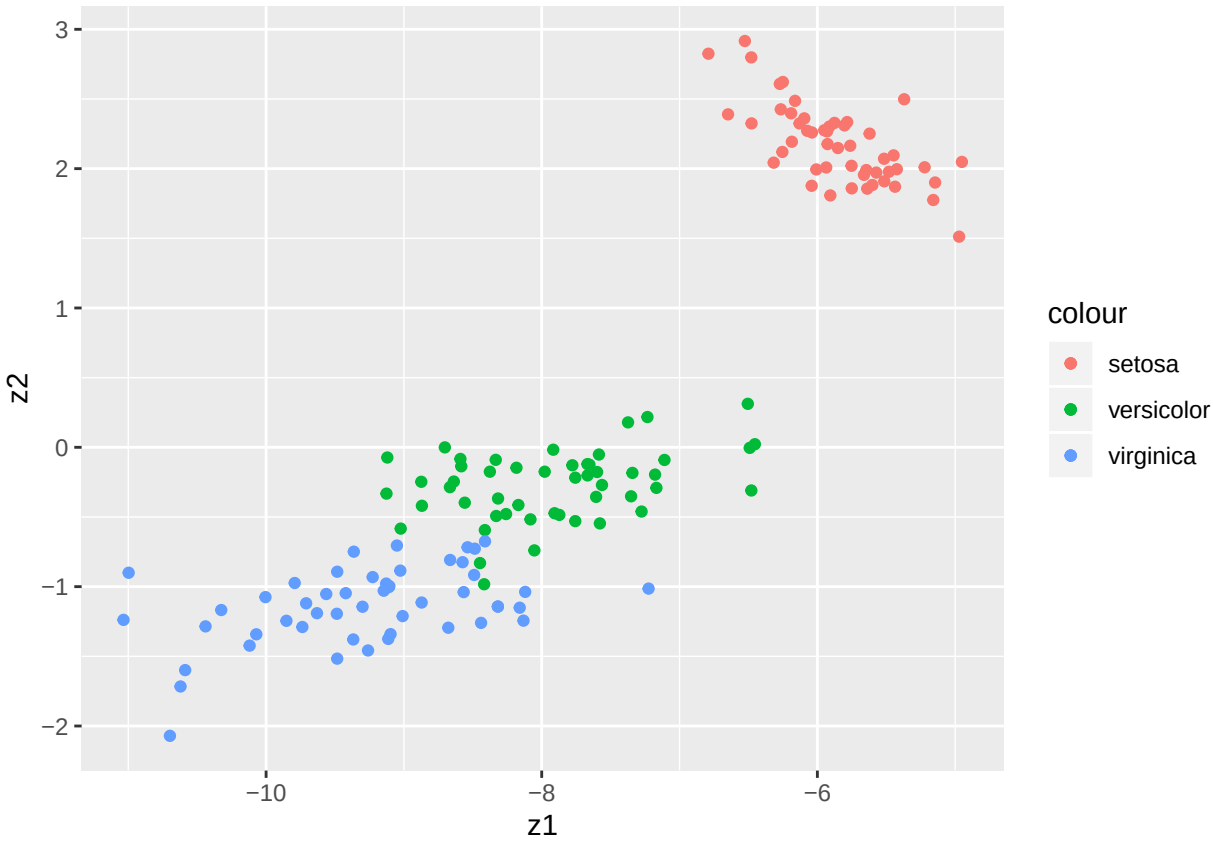
```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width
## [1,]            0           0            0           0
## [2,]            0           0            0           0
## [3,]            0           0            0           0
## [4,]            0           0            0           0
```

```r
C <- C(iris[ ,1:4], iris$Species)
```

```r
result <- eigen(t(C) %*% solve(Wiris) %*% C)
w <- result$vectors
u <- solve(Wiris) %*%C %*% w
z1 <- as.matrix(iris[,1:4]) %*% u[,1]
z2 <- as.matrix(iris[,1:4]) %*% u[,2]
plotdata <- data.frame("z1" = z1, "z2" = z2, group = iris$Species)
ggplot() +
  geom_point(aes(x = z1, y = z2, color = "setosa"), data = plotdata %>% filter(group == "setosa")) +
  geom_point(aes(x = z1, y = z2, color = "versicolor"), data = plotdata %>% filter(group == "versicolor
  geom_point(aes(x = z1, y = z2, color = "virginica"), data = plotdata %>% filter(group == "virginica"))
```




```
result <- eigen(t(as.matrix(iris[,1:4])) %*% as.matrix(iris[1:4]))
w <- result$vectors
z1 <- as.matrix(iris[,1:4]) %*% w[,1]
z2 <- as.matrix(iris[,1:4]) %*% w[,2]
plotdata <- data.frame("z1" = z1, "z2" = z2, group = iris$Species)
ggplot() +
  geom_point(aes(x = z1, y = z2, color = "setosa"), data = plotdata %>% filter(group == "setosa")) +
  geom_point(aes(x = z1, y = z2, color = "versicolor"), data = plotdata %>% filter(group == "versicolor
  geom_point(aes(x = z1, y = z2, color = "virginica"), data = plotdata %>% filter(group == "virginica")
```



The princical component projection offers us the largest total variance, regardless of the information of the group of variables. There're still significant overlapping in the axis of the first principal component that leads to a somehow difficult task to classify objects. CDA, on the other hand, offers us the best "classification" projection, taking the information of group into consideration.

```
distance <-  function(x, newx, Wiris){
  return((x - newx) %*% solve(Wiris) %*% t(x-newx))
}
x <- iris[,1:4]
group <- iris$Species
Wiris <- within_variance(x, group)
group <- factor(group)
centroid <- matrix(0, nrow = ncol(x), ncol = length(levels(group)))
k = 0
for (i in levels(group)){
  k = k+1
  centroid[,k] = apply(x[group == i, ],2,mean)
}
x1 =c(5.0,3.0,1.5,0.5)
x2 =c(5.5,3.0,6.0,2.0)
x3 =c(6.0,3.0,4.0,1.0)
x4 =c(5.0,3.0,1.0,0.5)
```

```
CDApred <- function(x, W = Wiris, centroid = centroid, group = iris$Species){
  dis = c()
  for (i in 1:length(levels(group))){
    dis = c(dis, distance(centroid[,i], t(x), W))
```

```
  }
  if (which.min(dis) == 1){print(paste("It shall be classified into setosa. The Mdistance is ", dis[1]))
  if (which.min(dis) == 2){print(paste("It shall be classified into versicolor. The Mdistance is ", dis
  if (which.min(dis) == 3){print(paste("It shall be classified into virginica. The Mdistance is ", dis[3
}

CDApred(x1, Wiris, centroid)
```

```
## [1] "It shall be classified into setosa. The Mdistance is  6.73571967792109"
```

```
CDApred(x2, Wiris, centroid)
```

```
## [1] "It shall be classified into virginica. The Mdistance is  25.7290782672213"
```

```
CDApred(x3, Wiris, centroid)
```

```
## [1] "It shall be classified into versicolor. The Mdistance is  5.28085878308332"
```

```
CDApred(x4, Wiris, centroid)
```

```
## [1] "It shall be classified into setosa. The Mdistance is  13.3102216730972"
```