419 Final Project

Justin Koida, PUT NAMES

Table of Contents

loading in stuff

Getting stuff running

REMOVE THIS BEFORE SUBMISSION

In order for people to run this, you will need to download some stuff as described below

First, you need to have quarto installed.

Once you have quarto, here are the required packages this project uses: tidyverse and knitr. You can install these by running the following commands in the R studio console

install.packages("tidyverse")

install.packages("knitr")

In order to render the pdf, you will need to download tinytex to your local device by running the following command in the terminal.

quarto install tinytex

Once all of this is done, should be good to go.

```r
library(tidyverse)
```

```r
discrim <- function(Y, group){
Y <- data.matrix(Y)
group <- as.factor(group)
m1 <- manova(Y ~ group)
nu.h <- summary(m1)$stats[1]
nu.e <- summary(m1)$stats[2]
p <- ncol(Y)
SS <- summary(m1)$SS
E.inv.H <- solve(SS$Residuals) %*% SS$group
eig <- eigen(E.inv.H)
s <- min(nu.h, p)
lambda <- Re(eig$values[1:s])
a <- Re(eig$vectors[,1:s])
a.star <- (sqrt(diag(SS$Residuals/nu.e)) * a)
return(list("a"=a, "a.stand"=a.star))
}

discr.sig <- function(Y, group){
Y <- data.matrix(Y)
group <- as.factor(group)
m1 <- manova(Y ~ group)
sums <- summary(m1)
evals <- sums$Eigenvalues
nu.e <- m1$df
nu.h <- m1$rank-1
k <- nu.h + 1
p <- ncol(m1$coef)
N <- nu.e + nu.h + 1
s <- min(p, nu.h)
lam <- numeric(s)
dfs <- numeric(s)
for(m in 1:s){
```

```
lam[m] <- prod(1/(1+evals[m:s]))
dfs[m] <- (p-m+1)*(k-m)
}
V <- -(N - 1 - .5*(p+k))*log(lam)
p.val <- 1 - pchisq(V, dfs)
out <- cbind(Lambda=lam, V, p.values=p.val)
dimnames(out)[[1]] <- paste("LD",1:s,sep="")
return(out)
}

partial.F <- function(Y, group){
Y <- data.matrix(Y)
group <- as.factor(group)
p <- ncol(Y)
m1 <- manova(Y ~ group)
nu.e <- m1$df
nu.h <- m1$rank-1
Lambda.p <- summary(m1,test="Wilks")$stats[3]
Lambda.p1 <- numeric(p)
for(i in 1:p){
dat <- data.matrix(Y[,-i])
m2 <- manova(dat ~ group)
Lambda.p1[i] <- summary(m2,test="Wilks")$stats[3]
}
Lambda <- Lambda.p / Lambda.p1
F.stat <- ((1 - Lambda) / Lambda) * ((nu.e - p + 1)/nu.h)
p.val <- 1 - pf(F.stat, nu.h, nu.e - p + 1)
out <- cbind(Lambda, F.stat, p.value = p.val)
dimnames(out)[[1]] <- dimnames(Y)[[2]]
ord <- rev(order(out[,2]))
return(out[ord,])
}

discr.plot <- function(Y, group, leg = NULL){
a <- discrim(Y, group)$a
z <- data.matrix(Y) %*% a
plot(z[,1], z[,2], type = "n", xlab = "LD1", ylab="LD2")
for(i in 1:length(unique(group))){
points(z[group == unique(group)[i],1],
z[group == unique(group)[i],2], pch = i)
```

```
}
#if(is.null(leg)) leg <- as.character(unique(group))
#legend(locator(1),legend = leg,pch=1:length(unique(group)))
}
```

```
pollution <- read_csv(here::here("pollution_419.csv"))
```

```
knitr::kable(pollution, caption = "Pollution")
```

Table 1: Pollution

MORTRANK	PRECIP	EDUC	NONWHITE	NOX	SO2
1	13	12.2	3.0	32	3
1	28	12.1	7.5	2	1
1	10	12.1	5.9	66	20
1	43	9.5	2.9	7	32
1	25	12.1	2.0	11	26
1	35	11.8	14.8	1	1
1	60	11.5	13.5	1	1
1	11	12.1	7.8	319	130
1	31	10.9	5.1	3	10
1	15	12.2	4.7	8	28
1	32	11.1	5.0	4	18
1	43	11.5	7.2	3	10
1	31	11.4	11.5	1	1
1	37	12.0	3.6	21	44
1	45	11.1	1.0	3	8
1	35	12.2	5.7	7	20
1	45	10.6	5.3	4	4
1	45	11.1	3.4	4	20
1	18	12.2	13.7	171	86
1	42	9.0	4.8	8	49
2	40	10.3	2.5	2	11
2	36	10.7	6.7	7	20
2	35	12.0	12.6	4	4
2	36	11.4	8.8	15	59
2	46	11.3	8.8	3	8
2	30	11.1	5.8	23	125

MORTRANK	PRECIP	EDUC	NONWHITE	NOX	SO2
2	43	12.1	3.5	32	62
2	36	11.4	12.4	4	16
2	42	10.1	2.2	4	18
2	30	10.8	13.1	4	11
2	41	9.6	2.7	11	89
2	38	11.4	3.8	5	25
2	46	11.4	21.0	5	1
2	34	9.7	17.2	15	68
2	38	10.7	11.7	13	39
2	37	11.9	13.1	9	15
2	31	10.8	15.8	35	124
2	45	10.1	21.0	14	78
2	44	9.8	0.8	6	33
2	41	12.3	25.9	28	102
3	39	11.4	15.6	7	33
3	40	10.2	13.0	26	146
3	42	10.4	22.7	3	5
3	31	10.7	9.5	7	25
3	47	11.1	27.1	8	24
3	35	11.1	14.7	21	64
3	30	9.9	13.1	37	193
3	36	10.6	8.1	59	263
3	42	10.7	11.3	26	108
3	35	11.0	3.5	10	39
3	36	10.5	8.1	12	37
3	45	11.3	12.1	11	42
3	50	10.4	36.7	18	34
3	42	10.5	17.5	32	161
3	52	9.6	22.2	8	27
3	33	10.9	16.3	63	278
3	44	11.0	28.6	9	48
3	53	10.2	38.5	32	72
3	43	9.6	24.4	38	206
3	54	9.7	31.4	17	1

Section A

Section B

Section C

C.1

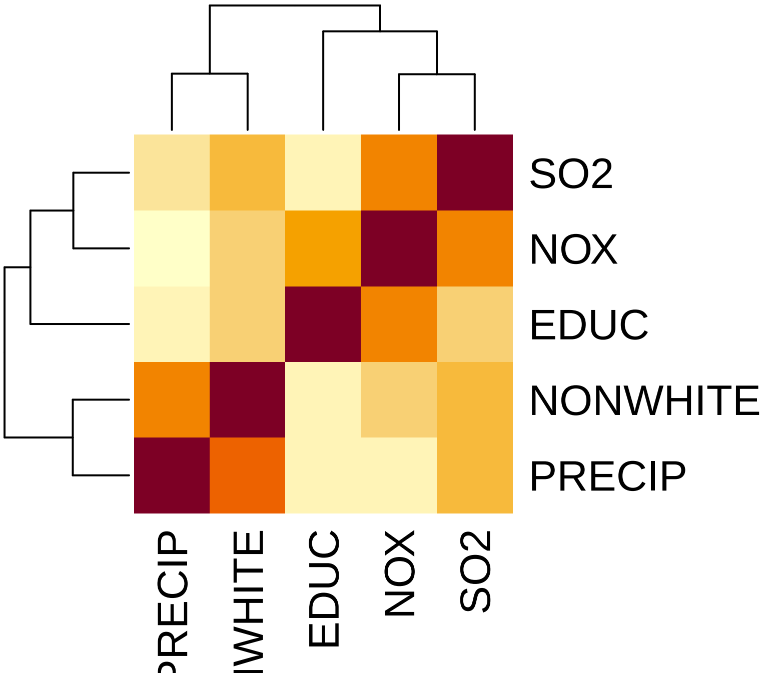
```
cor_matrix <- round(cor(pollution[,-1]), 6)
knitr::kable(cor_matrix, caption = "Correlation Matrix of Pollution Variables")
```

Table 2: Correlation Matrix of Pollution Variables

	PRECIP	EDUC	NONWHITE	NOX	SO2
PRECIP	1.000000	-0.490425	0.413204	-0.487321	-0.106924
EDUC	-0.490425	1.000000	-0.208774	0.224402	-0.234346
NONWHITE	0.413204	-0.208774	1.000000	0.018385	0.159293
NOX	-0.487321	0.224402	0.018385	1.000000	0.409394
SO2	-0.106924	-0.234346	0.159293	0.409394	1.000000

The most correlation we see between variables is EDUC and PRECIP with a correlation of -.4904. This is not correlated significantly, so we will keep all the variables in this data set.

```
heatmap(cor_matrix)
```



C.2

C.2 Part 1

Write the complete form of all discriminant functions. Be sure to use standardized coefficients. Based on these coefficients, produce a ranking of variable importance (in the presence of other variables).

To start, we have k = 3 groups and p = 6 variables, so we have min(p, k-1) = 2 discriminant functions.

```
discrim(pollution[,-1], pollution[[1]])
```

```
$a
            [,1]        [,2]
[1,] -0.01840701 -0.54309519
[2,]  0.97027032  0.40472511
[3,] -0.23875696  0.70885075
[4,]  0.01988270  0.19533883
[5,] -0.02892210 -0.02487075
```

```
$a.stand
           [,1]       [,2]
[1,] -0.1722439 -5.0820234
[2,]  0.7507216  0.3131456
[3,] -1.7675389  5.2476849
[4,]  0.9193970  9.0326716
[5,] -1.6834803 -1.4476618
```

Here is the complete form of both discriminant functions:

LD1 = -0.1722439y1 + 0.7507216y2 - 1.7675389y3 + 0.9193970y4 - 1.6834803y5

LD2 = -5.0820234y1 + 0.3131456y2 + 5.2476849y3 + 9.0326716y4 - 1.4476618y5

For the importance of variables based on the standardized coefficients, we have

LD1: 3, 5, 4, 2, 1 –> NONWHITE, SO2, NOX, EDUC, PRECIP

LD2: 4, 3, 1, 5, 2 –> NOX, NONWHITE, PRECIP, SO2, EDUC

C.2 Part 2

Carry out tests of significance for the discriminant functions. Be sure to specify corresponding null and alternative hypotheses, test statistic values, and p-values. Be sure to provide a conclusion for each test.

```
discr.sig(pollution[,-1], pollution[[1]])
```

```
       Lambda         V     p.values
LD1 0.4375897 45.456049 1.799089e-06
LD2 0.9616608  2.150142 7.081668e-01
```

let α1, α2, …, αs be the population discriminant functions

For LD1,

H0: α1 = α2 = 0 <——- where 0 is the 0 vector

Ha: at least one α1 or α2 != 0

α* = .05/2 = .025

With a p-value of 1.799089e-06, we reject the Null Hypothesis at α* = .025, so we conclude that we have significant evidence that at least one α1, α2 != 0 at the α* = .025 level.

For LD2,

H0: α2 = 0 <——- where 0 is the 0 vector

Ha: α2 != 0

α* = .05/2 = .025

With a p-value of 7.081668e-01, we do not reject the Null Hypothesis at α* = .025, so we do not have significant evidence to suggest that α2, the second linear discriminant function, has a significant impact on group separability.

Conclude:

We have sufficient evidence to conclude α1 significantly contributes to group separability at α* = .025. We have insufficient evidence to conclude α2 significantly contributes to group separability at α* = .025.

C.2 Part 3

Carry out tests of significance of each non-grouping variable, after adjusting the presence of other non-grouping variables. Be sure to specify corresponding null and alternative hypotheses, test statistic values, and p-values. Be sure to provide a conclusion for each test.

```
partial.F(pollution[,-1], pollution[[1]])
```

```
            Lambda    F.stat      p.value
NONWHITE 0.7351297 9.5480596 0.0002875051
SO2      0.8028967 6.5054908 0.0029749625
NOX      0.9394342 1.7084679 0.1909672716
EDUC     0.9559307 1.2216740 0.3029006436
PRECIP   0.9948165 0.1380797 0.8713416049
```

Using α* = .05/5 = .01

We will check variable importance for each non-grouping variable after adjusting for the presence of the other non-grouping variables.

NONWHITE

H0: The variable NONWHITE has no significant effect on group separation

Ha: The variable NONWHITE has a significant effect on group separation

With a Lambda stat of 0.7351297 and a p-value of 0.0002875051, we reject H0 at α* = .01, so there is strong evidence that NONWHITE individually contributes significantly to group separability, adjusting for other variables.

SO2

H0: The variable SO2 has no significant effect on group separation

Ha: The variable SO2 has a significant effect on group separation

With a Lambda stat of 0.8028967 and a p-value of 0.0029749625, we reject H0 at α* = .01, so there is strong evidence that SO2 individually contributes significantly to group separability, adjusting for other variables.

NOX

H0: The variable NOX has no significant effect on group separation

Ha: The variable NOX has a significant effect on group separation

With a Lambda stat of 0.9394342 and a p-value of 0.1909672716, we do not reject H0 at α* = .01, and so there is insufficient evidence to conclude that NOX, individually, contributes significantly to group separation adjusting for other variables.

EDUC

H0: The variable EDUC has no significant effect on group separation

Ha: The variable EDUC has a significant effect on group separation

With a Lambda stat of 0.9559307 and a p-value of 0.3029006436, we do not reject H0 at α* = .01, and so there is insufficient evidence to conclude that EDUC, individually, contributes significantly to group separation adjusting for other variables.

PRECIP

H0: The variable PRECIP has no significant effect on group separation

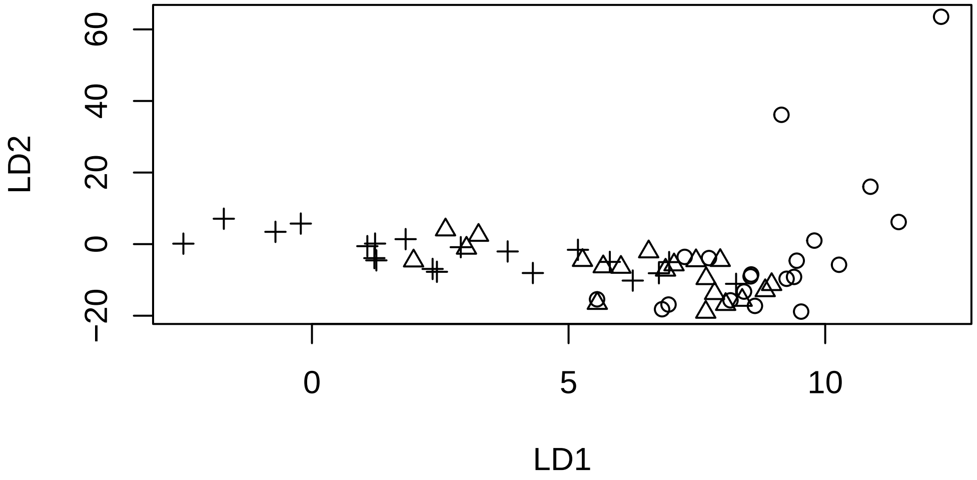
Ha: The variable PRECIP has a significant effect on group separation

With a Lambda stat of 0.9948165 and a p-value of 0.8713416049, we do not reject H0 at α* = .01, and so there is insufficient evidence to conclude that PRECIP, individually, contributes significantly to group separation adjusting for other variables.

C.2 Part 4

Produce a plot of the first two linear discriminant functions. Be sure to include this plot in the report. Comment on the plot with regard to how well the discriminant functions separate the groups in the data. NOTE: When using the corresponding function, the system will wait for you to click on the graph to place a legend for the symbols used in the graph. Be sure to click on a location that is empty for the legend placement. If the legend placement causes your system to lock up, modify the original function and remove the code that inserts the legend. In that event, you can manually insert your own legend by annotating the graph

```
discr.plot(pollution[,-1], pollution[[1]])
```



LD1 seems to separate a decent amount of the pluses and circles when looking at the projection onto LD1. However, there still seems to be some overlap between a few of the plues, a few of the circles, and many of the triangles.

LD2 does not separate the groups well at all.