## Contents

Preliminaries
0: Introduction
1: Graphical Displays
2: Reduction of Dimensionality
3: Multidimensional Scaling Techniques
**4: Discriminant Analysis**
5: Multivariate Regression Analysis
6: Canonical Correlation Analysis
7: Partial Least Squares
8: Statistical Analysis of Multivariate Data
9: Statistical Discriminant Analysis



## Discriminant Analysis

- **Techniques for when data classified into *known* groups**
  - Objectives:-
    - Effective data display utilising this extra information
    - Dimensionality reduction keeping information on group differences
    - assessment of group differences by examination of loadings (c.f. PCA)
    - Classification of new observations into the groups



- **Methods:-**
  - **L**inear **D**iscriminant **A**nalysis
    - Based upon linear combinations of variables
    - Helps with all objectives
      - (cf PCA)
    - Can assess effectiveness of classification by simulation, jackknifing, randomization etc, (App 1)
    - **LDA** is a ***data analytic*** method
      - based just upon the data
    - (See Chapter 9 for Statistical Discrimination
      - Based on statistical models)
    - **LDA** is a ***supervised learning technique***



- If **primary objective** is *classification* of new observations other techniques may be better in **some** cases
  - Quadratic Discriminant Analysis
    - See appendix 2
  - Neural Networks
    - See appendix 8
  - k-means classification
  - Nearest neighbour methods
  - ……&c.,&c.,….



- **Setup:-**
  - k ***known*** groups or categories $G_i$; i=1,…k
  - $n_i$ observations from each group, $\Sigma n_i = n$
  - Each observation is p-dimensional
    - so data are $\{x_{ij}; i=1,...,k; j=1,...,n_i\}$, $x_{ij}$ a p×1 vector)
    - data matrix for $i^{th}$ group $G_i$ is $X_i'$ (so $X_i'$ is $n_i \times p$)
    - $S_i$ is the within group $G_i$ variance matrix (p×p)
    - **W** is the ***within groups variance***
    - **B** is the ***between groups variance***
      - **(see notes for formulæ)**



- **W** measures variability **inside** the groups
- **B** measures how **different** groups are
- Need to *compare* B with W
- Both W and B are p×p matrices
  - i.e. variability measures are in p-dimensions
- Want to reduce the dimensionality
  - *Retaining* information on **B** ←→ **W** comparison





- **Outline**
  - measure separation between known groups:- **$W^{-1}B$**
  - 'compares' B in relation to W
  - In p=1 dimension this is F = B / W
    - The F-ratio in ANOVA
    - If F is big then more separation between groups (when compared with internal variability)
  - In p dimensions must asses how 'big' $W^{-1}B$ is
    - This is a p×p matrix — difficult, possibilities are determinant, largest eigenvalue, …. (see later)



- **Outline (ctd.)**
  Idea is project data into one dimension
  - (i.e. find linear combination of variables)

  Which maximizes the F-statistic
  - (i.e. highlights distinction between groups)
- Theorem shews this is achieved by first eigenvector of $W^{-1}B$
  - i.e. corresponding to largest eigenvalue
    - strictly the **right** eigenvector ($W^{-1}B$ not symmetric)



- **Outline (ctd.)**
  - This is similar to the PCA theorem
    - (proof relies on **same procedure** as last time)
  - So, import analogies from PCA
    - Interpretation of coefficients of eigenvectors as loadings
    - Look also at 2nd,3rd,… eigenvectors of $W^{-1}B$
    - Scree plots of eigenvalues
    - Plot data on eigenvectors (or **Crimcoords**)
      - even though eigenvectors are not orthogonal

**(end of outline)**



## A Procedure for Maximization

- 1: Introduce some constraint
- 2: Introduce a Lagrange multiplier & define a new objective function
- 3: Differentiate w.r.t. x and set =0
- 4: Recognise this is an eigenequation with the Lagrange multiplier as eigenvalue
- 5: Deduce that there are ONLY a limited number of possible values for this eigenvalue (all of which can be calculated numerically)
- 6: Use some extra step to determine which eigenvalue gives the maximum (typically use the constraint somewhere)



- **Derivation of Crimcoords**
  - Project data by vector $a_1$
    - i.e. take line combination of variables given by $a_1$
  - so data sets for the k groups are $X_i'a_1$
  - between groups variance is $\mathbf{a_1'Ba_1}$
  - within groups variance is $\mathbf{a_1'Wa_1}$
  - $W^{-1}B$ becomes

$$F_1 = \frac{a_1'Ba_1}{a_1'Wa_1}$$

  - So need to choose $a_1$ to maximize $F_1$



- Note that if we multiply $a_1$ by a scalar constant c then $F_1$ does not alter

$$F_1 = \frac{a_1'Ba_1}{a_1'Wa_1} = \frac{(ca_1)'B(ca_1)}{(ca_1)'W(ca_1)}$$

for any scalar c

  - So can impose a ***scale constraint*** on $a_1$
    - Could take $a_1'a_1 = 1$
    - Simpler to take $a_1'Wa_1 = 1$
      - (makes differentiating $F_1$ easier)
    - Maximize $F_1$ subject to constraint $a_1'Wa_1 = 1$
    - i.e. maximize $a_1'Ba_1$ subject to constraint $a_1'Wa_1 = 1$





- Introduce a Lagrange multiplier $\lambda_1$:
  let $\Omega_1 = a_1'Ba_1 - \lambda_1(a_1'Wa_1 - 1)$

  so $\frac{\partial\Omega_1}{\partial a_1} = 2Ba_1 - 2\lambda_1 Wa_1 = 0$
- i.e. $W^{-1}Ba_1 - \lambda_1 a_1 = 0$
- i.e. $a_1$ is eigenvector of $W^{-1}B$ corresponding to $\lambda_1$
  - strictly a *right* eigenvector since $W^{-1}B$ not symmetric



- Which eigenvalue is $\lambda_1$?

  $W^{-1}Ba_1 - \lambda_1 a_1 = 0$
  - pre-multiply by $a_1'W$

  so $a_1'Ba_1 = \lambda_1 a_1'Wa_1 = \lambda_1$
  - (since $a_1'Wa_1 = 1$ — the original constraint)
  - **SO** to maximize $a_1'Ba_1$ with $a_1'Wa_1 = 1$ take $\lambda_1$ as the *largest* eigenvalue of $W^{-1}B$ & $a_1$ as the corresponding eigenvector
    - (aside: compare Exercises 1, Q3)

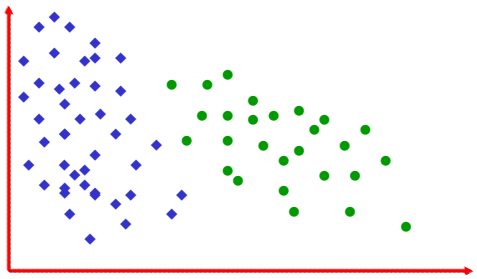


- **Fisher's Linear Discriminant Function**
  - The function of the p-vector x

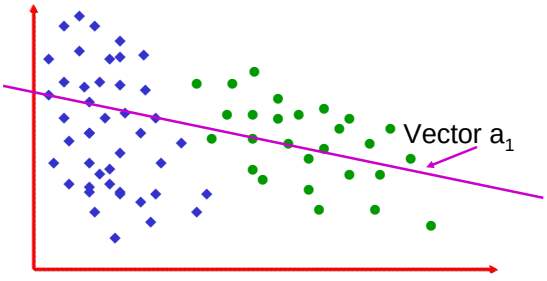
    **$f(x) = a_1'x$**

    is known as *Fisher's linear discriminant function*
  - The value of this is the most sensitive to which group or category x belongs
  - to classify a new observation x a rule based on the value of f(x) is a leading contender
    - (but not the only contender)
      - (see later)

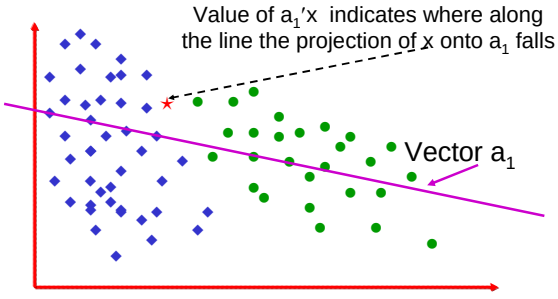


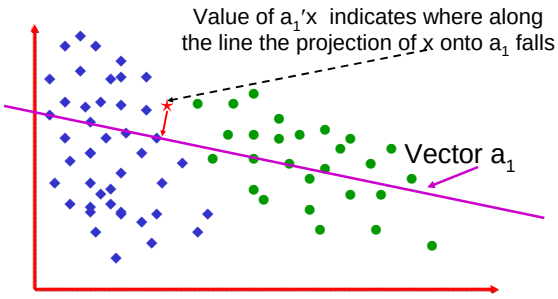




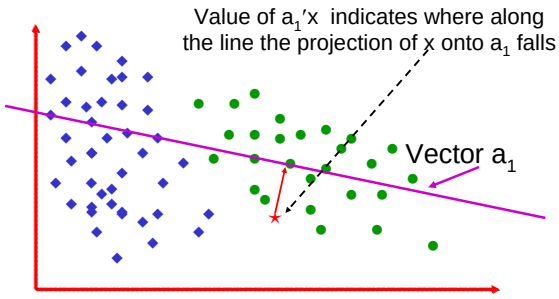











- ? Other eigenvalues of $W^{-1}B$ ?
  - $W^{-1}B$ may have non-zero eigenvalues $\lambda_1,\lambda_2,...\lambda_r$
    - where $r = \text{rank}(W^{-1}B) \le \min(k-1,p)$
    - ( = unless collinearities in group means)
    - (or in data points inside each group)
      - or linear dependence between variables
  - Only $\lambda_1$ and $a_1$ have strict interpretations
  - What can we do with the other eigenvalues and eigenvectors?



- Note eigenvectors of $W^{-1}B$ non-orthogonal
  - $W^{-1}B$ is not symmetric (though both W and B are)
  - However (for $i^{th}$ & $j^{th}$):

$$W^{-1}Ba_i - \lambda_i a_i = 0 \Rightarrow Ba_i - \lambda_i Wa_i = 0$$
$$\& \; W^{-1}Ba_j - \lambda_j a_j = 0 \Rightarrow Ba_j - \lambda_j Wa_j = 0$$
$$\text{so } a_j'Ba_i - \lambda_i a_j'Wa_i = 0$$
$$\& \; a_i'Ba_j - \lambda_j a_i'Wa_j = 0$$

  - W & B symmetric;
    - so $a_j'Ba_i = a_i'Ba_j$ & $a_j'Wa_i = a_i'Wa_j$ and $\lambda_i \ne \lambda_j$

**so $a_j'Wa_i = 0$ for $i \ne j$**



- **$a_j'Wa_i = 0$ for $i \ne j$**
  - $a_i$ & $a_j$ are 'orthogonal wrt W'
  - The within groups variance W reflects the different variances and covariances of the variables
  - If $W = \sigma^2 I_p$
    - (i.e. all variables independent variance $\sigma^2$)

    then **$a_j'a_i = 0$** and $a_i$ & $a_j$ are orthogonal
  - If W=D=diag($\sigma^2_{kk}$) [a diagonal matrix]
    - (i.e. all variables independent, different variances)

    then **$a_j'a_i = \Sigma\sigma^2_{kk}a_{ik}a_{jk} = 0$**
  - W 'corrects' for the variance structure of the variables



- **Definitions:**
  - The functions $a_i'x$, i=1,2,...,r are called the ***discriminant coordinates*** or ***crimcoords***
  - The space spanned by [the first t of] them is called the ***discriminant space***
  - choose suitable t by scree plot of $\lambda_1,\lambda_2,...$
    - (by analogy with PCA)
    - but don't have 'amount of discrimination' partitioned into 'separate bits'
      - contradistinction to PCA





- Traditional to plot crimcoords as if they were on orthogonal axes
- This 'distorts' the original space
  - but in a sensible way
  - equivalent to a form of 'average standardization' of the variables to bring them to a common scale of measurement
    - (i.e. common st.devs and 'orthogonalized')
      - Note:- common ≡ ***within groups*** (averaged over groups)



  - transforming to discriminant space appropriate for displaying relationships & distinctions between groups
    - (allowing for the different scales of the variables and their structure)
  - Given this, examining loadings of variables in crimcoords indicates which variables provide discrimination between groups
    - c.f. interpretation of principal components



- **Computation**
  - S-plus:- function `lda(.)` in MASS library
    - full facility: `help(lda)` to find out more
  - Minitab:
    - **Stat>Multivariate>Discriminant Analysis**
      **BUT cannot produce plots on crimcoords**
    - **Can** produce crimcoords 'by hand' from command line using spectral decomposition of W & eigenanalysis of B
      - Cumbersome
    - Or use a ***slick trick:***
      - cut&paste from **MANOVA** output



- Implementation in **R**
  - Use command line function `lda()` in MASS library
    - **M**odern **A**pplied **S**tatistics with **S**
      - By Venables & Ripley
  - syntax: `lda(group~`$V_1+V_2+\ldots+V_p$`)`



- syntax: `lda(group~`$V_1+V_2+\ldots+V_p$`)`
- Coordinates on crimcoords obtained in `predict.lda(lda(group~`$V_1+V_2+\ldots+V_p$`))$x`
- i.e. data matrix multiplied by matrix of eigenvectors of $W^{-1}B$
  - can then plot this to obtain plot on crimcoords
- To get new data multiplied by this matrix: `predict.lda(lda(group~`$V_1+V_2+\ldots+V_p$`),newdata)$x`
- Gives new data matrix multiplied by this
  - best to store objects at each stage



- Example:- Anderson's Iris Data
  - Historical note:-
    - Edgar Anderson consulted Ronald Fisher about the problem which motivated Fisher to develop the methodology of LDA ~70 yrs b.p.
    - most over-analyzed data set in world
    - if a technique doesn't work on the Iris Data then technique is 'useless'





```
> attach(irisnf)
> library(MASS)
> iris.lda<-
  lda(Variety~Sepal.l+Sepal.w+Petal.l+Petal.w)
> iris.crimcoord<-predict.lda(iris.lda)$x
> iris.lda
> plot(iris.lda)
> plot(iris.crimcoord)
```



`> attach(irisnf)`

- Attach data set

`> library(MASS)`

- Attach library containing `lda(.)` function

```
> iris.lda<-
+ lda(Variety~Sepal.l+Sepal.w+Petal.l+Petal.w)
```

- Perform **l**inear **d**iscriminant **a**nalysis and store in `iris.lda`

`> iris.crimcoords<-predict.lda(iris.lda)$x`

- Store coordinates in `iris.crimcoords`

`> iris.lda`

- Print everything in object iris.lda



`> iris.lda` ← print everything in object iris.lda

```
Call:
lda(Variety ~ Sepal.l + Sepal.w + Petal.l + Petal.w)

Prior probabilities of groups:
        1         2         3
0.3333333 0.3333333 0.3333333

Group means:
  Sepal.l Sepal.w Petal.l Petal.w
1   5.006   3.428   1.462   0.246
2   5.936   2.764   4.260   1.326
3   6.588   2.974   5.552   2.026

Coefficients of linear discriminants:
              LD1         LD2
Sepal.l -0.8129912  0.006189401
Sepal.w -1.5569539  2.157270079
Petal.l  2.1987600 -0.919346099
Petal.w  2.8266164  2.825877950

Proportion of trace:
    LD1    LD2
0.9912 0.0088
```

Eigenvectors of $W^{-1}B$ corresponding to non-zero eigenvalues: elements give loadings of variables in each crimcoord.

Eigenvalues as a percentage of total



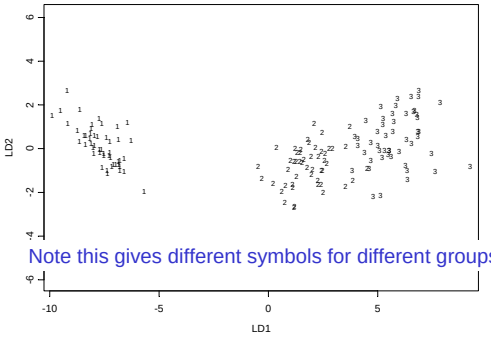
```
Coefficients of linear discriminants:
              LD1         LD2
Sepal.l -0.8129912  0.006189401
Sepal.w -1.5569539  2.157270079
Petal.l  2.1987600 -0.919346099
Petal.w  2.8266164  2.825877950

Proportion of trace:
    LD1    LD2
0.9912 0.0088
```

Numerically different from values produced by Minitab but only by a scale factor ⇒ same interpretation

```
  Eigenvector  1         2
  sepal-l    0.0671   -0.0005
  sepal-w    0.1284   -0.1779
  petal-l   -0.1814    0.0758
  petal-w   -0.2331   -0.2331
```



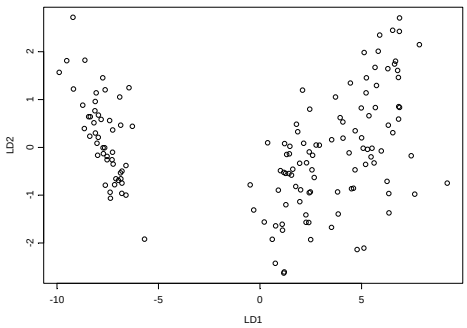
`> plot(iris.lda)`




Note this gives different symbols for different groups …



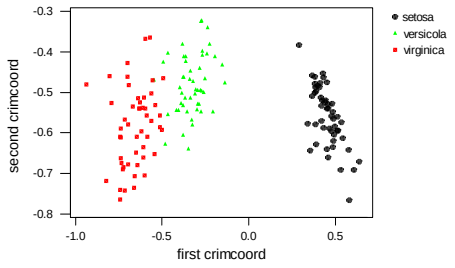
`> plot(iris.crimcoord)`




…and this does not give different symbols for different groups





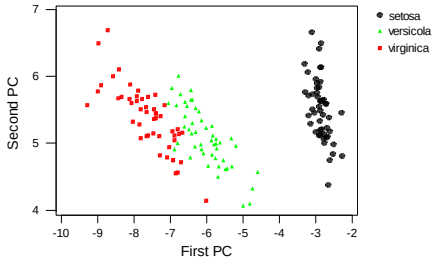



Plot of iris data on crimcoords



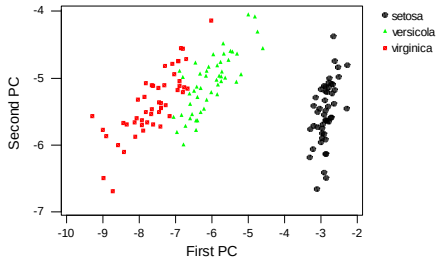
- **Slightly better separation than plots on principal components**
- **Note that signs of principal components and crimcoords are arbitrary**
- **Can flip plot to mirror image by**
  - `MTB > let 'second PC' = - 'second PC'`





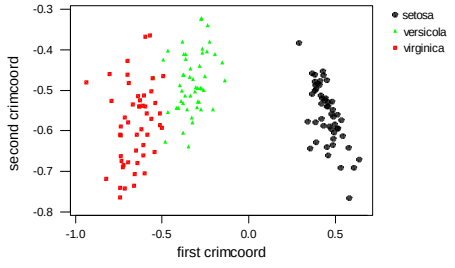

Plot of iris data on princoords





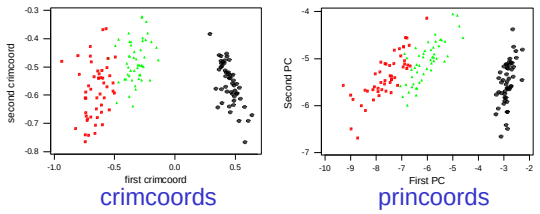

Plot of iris data on princoords






Plot of iris data on crimcoords





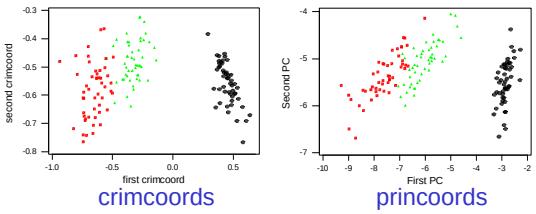

crimcoords princoords

Slightly better separation between groups on crimcoords than on principal components

More noticeable differences on less 'ideal data' than Anderson's Iris Data (see e.g. Exercises 2)



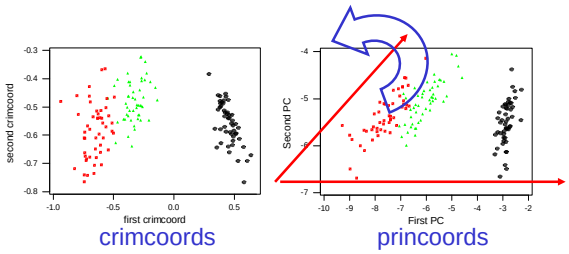





crimcoords princoords

Note 'distortion' of data on crimcoords: as if true axes were at acute angle and have been stretched round

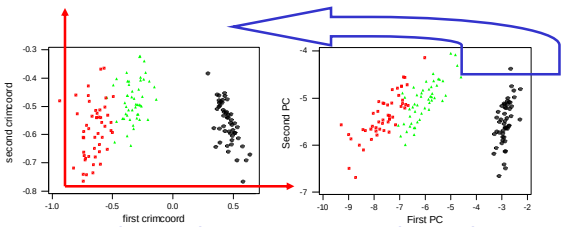





crimcoords princoords

'Pulling out' the oblique axis to stretch the data space gives a plot similar to crimcoords






crimcoords princoords

And then the plot on crimcoords: has the 1st crimcoord orthogonal to the direction of maximal separation between the groups.



- Minitab:
  - MANOVA: Multivariate Analysis of Variance
    - for statistical testing of differences between groups
      - (see next chapter)
    - option available for producing eigenvectors (& values) of $W^{-1}B$
  - **ANOVA>Balanced MANOVA...**
    - followed by checking the **'Eigen analysis'** box on the **Results…** menu



- **ANOVA>Balanced MANOVA...**
  - followed by checking the **'Eigen analysis'** box on the **Results…** menu
- produces the commands for the eigenanalysis of the $W^{-1}B$

```
MTB > ANOVA 'Sepal-l'-'Petal-w' = Variety;
SUBC>   MANOVA;
SUBC>   Eigen;
SUBC>   NoUnivariate.
```



- **ANOVA>Balanced MANOVA...**
  - followed by checking the **'Eigen analysis'** box on the **Results…** menu
- produces the commands for the eigenanalysis of the $W^{-1}B$

```
MTB > ANOVA 'Sepal-l'-'Petal-w' = Variety;
SUBC>   MANOVA;
SUBC>   Eigen;
SUBC>   NoUnivariate.
```





- **Results:**

```
Eigenvalue 32.4704    0.2883    0.0000    0.0000
Proportion  0.9912    0.0088    0.0000    0.0000
Cumulative  0.9912    1.0000    1.0000    1.0000

Eigenvector  1         2         3         4
sepal-l    0.0671   -0.0005    0.2173    0.1481
sepal-w    0.1284   -0.1779   -0.2171    0.0154
petal-l   -0.1814    0.0758   -0.2461    0.0329
petal-w   -0.2331   -0.2331    0.3170   -0.2033
```



- **Results:**

```
Eigenvalue 32.4704    0.2883    0.0000    0.0000
Proportion  0.9912    0.0088    0.0000    0.0000
Cumulative  0.9912    1.0000    1.0000    1.0000

Eigenvector  1         2         3         4
sepal-l    0.0671   -0.0005    0.2173    0.1481
sepal-w    0.1284   -0.1779   -0.2171    0.0154
petal-l   -0.1814    0.0758   -0.2461    0.0329
petal-w   -0.2331   -0.2331    0.3170   -0.2033
```

Zero eigenvalues mean eigenvectors arbitrary, non-interpretable, irrelevant



- **Results:**

```
Eigenvalue 32.4704    0.2883    0.0000    0.0000
Proportion  0.9912    0.0088    0.0000    0.0000
Cumulative  0.9912    1.0000    1.0000    1.0000

Eigenvector  1         2         3         4
sepal-l    0.0671   -0.0005    0.2173    0.1481
sepal-w    0.1284   -0.1779   -0.2171    0.0154
petal-l   -0.1814    0.0758   -0.2461    0.0329
petal-w   -0.2331   -0.2331    0.3170   -0.2033
```

Zero eigenvalues mean eigenvectors arbitrary, non-interpretable, irrelevant



- **Interpretation:**

```
Eigenvalue 32.4704    0.2883
Proportion  0.9912    0.0088
Cumulative  0.9912    1.0000
```

1st eigenvalue proportionally 0.99 shews most discrimination in just one component
(although 3 groups ⇒ 2 crimcoords)



- **Interpretation:**

```
Eigenvector  1         2
sepal-l    0.0671   -0.0005
sepal-w    0.1284   -0.1779
petal-l   -0.1814    0.0758
petal-w   -0.2331   -0.2331
```

**The first eigenvector is (0.07, 0.13, –0.18, –0.23)'.**
**So score on the first crimcoord is**
**0.07×sepal−l + 0.13×sepal−w**
**– 0.18×petal−l –0.23×petal−w**



- **Interpretation:**

```
Eigenvector  1         2
sepal-l    0.0671   -0.0005
sepal-w    0.1284   -0.1779
petal-l   -0.1814    0.0758
petal-w   -0.2331   -0.2331
```

Positive coeffs for sepals
Negative coeffs for petals
⇒ Main distinction is sepal size *vs* petal size





- **Producing plots on crimcoords:**
  - Need to multiply data matrix by matrix of eigenvectors

$$\begin{pmatrix} 0.0671 & -0.0005 \\ 0.1284 & -0.1779 \\ -0.1814 & 0.0758 \\ -0.2331 & -0.2331 \end{pmatrix}$$



- To multiply data by eigenvectors need to copy columns into matrices
  - Use Calc > Matrices > Copy to generate

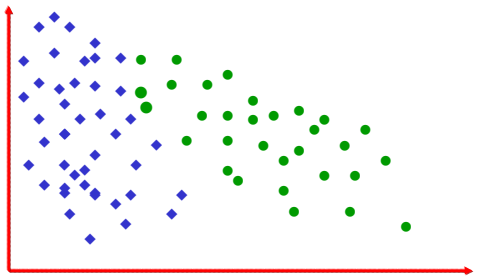
```
MTB > Copy 'Sepal-l'-'Petal-w' m1.
  Cut and paste eigenvectors
  to C8 and C9 from session window
MTB > Copy  C8 C9 m2.
MTB > Multiply m1 m2 m3.
MTB > Copy m3 c50 c51.
```

Then C50 and C51 have the data transformed to 1st two crimcoords

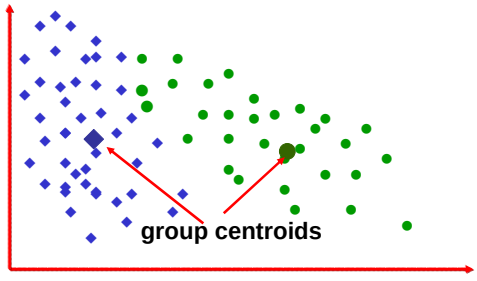


- **Application to Informal Classification**
  - (i.e. Data Analytic Classification)
  - k groups of 'reference' data $X_i$'; j = 1,…,k
  - set of r unknown points $u_j$; j = 1,...,r
  - allocate each $u_i$ to the 'closest' group

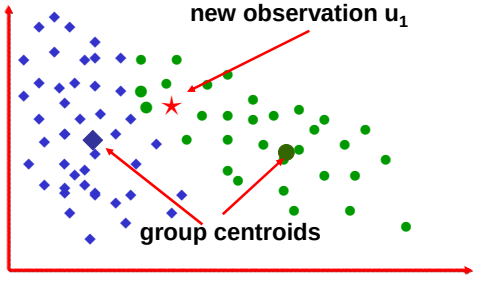


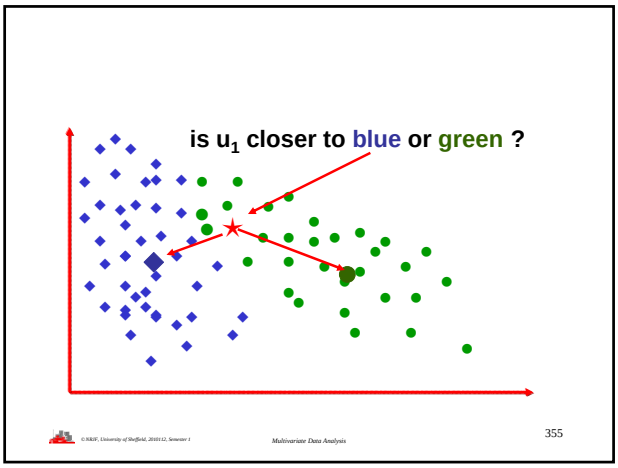


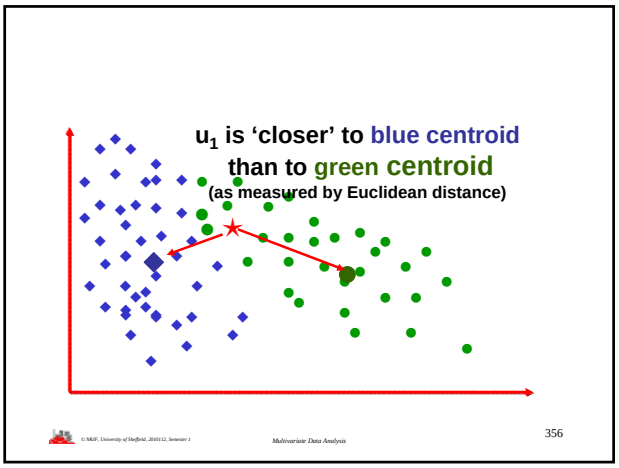


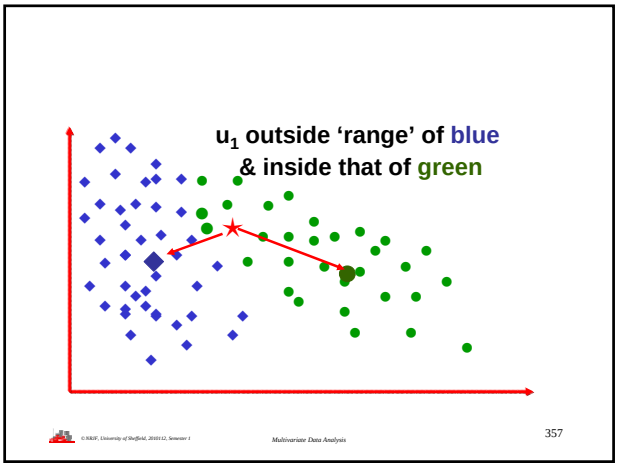


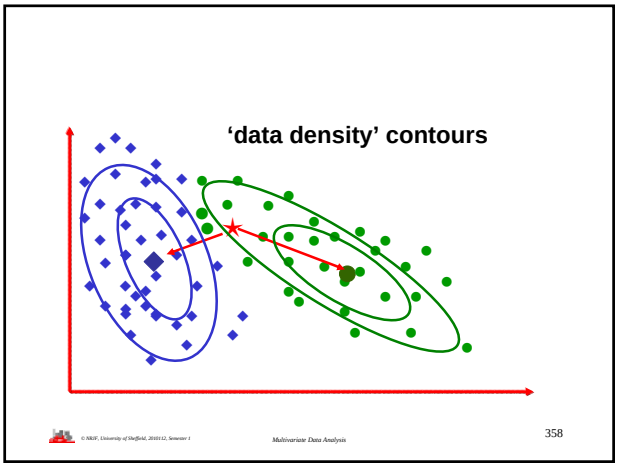


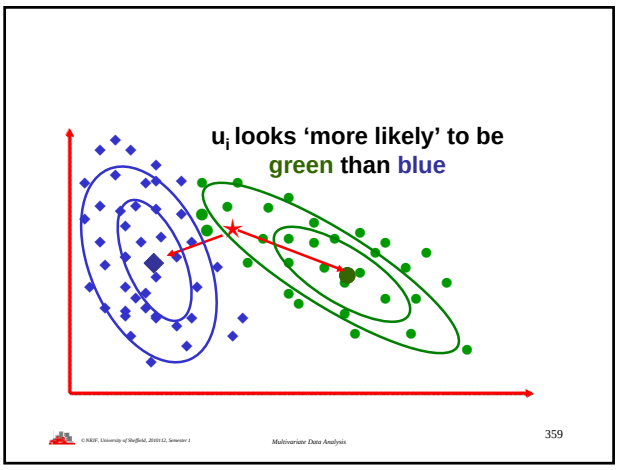



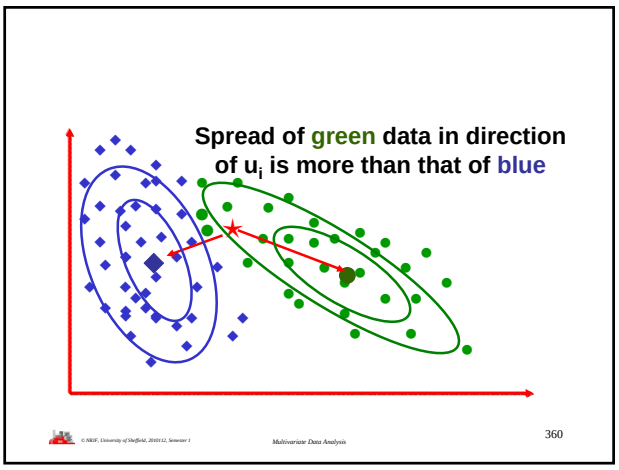

is $u_1$ closer to blue or green ?
© NRJF, University of Sheffield, 2010/12, Semester 1
Multivariate Data Analysis
355


$u_1$ is 'closer' to blue centroid
than to green centroid
(as measured by Euclidean distance)
© NRJF, University of Sheffield, 2010/12, Semester 1
Multivariate Data Analysis
356


$u_1$ outside 'range' of blue
& inside that of green
© NRJF, University of Sheffield, 2010/12, Semester 1
Multivariate Data Analysis
357


'data density' contours
© NRJF, University of Sheffield, 2010/12, Semester 1
Multivariate Data Analysis
358


$u_i$ looks 'more likely' to be
green than blue
© NRJF, University of Sheffield, 2010/12, Semester 1
Multivariate Data Analysis
359


Spread of green data in direction
of $u_i$ is more than that of blue
© NRJF, University of Sheffield, 2010/12, Semester 1
Multivariate Data Analysis
360



- **Problem arises since extent of natural variation is different in different directions**
  - Sample space is not ***isotropic***
  - So, need to take differences in direction of variation into account when assessing distance of new observations from group centroids



- Could standardize the data in an analogous way to dividing by standard deviation in univariate data
- Equivalent to recording distance by

$$D^2(i) = (u - \bar{x}_i)' S_i^{-1} (u - \bar{x}_i)$$

if we standardize with respect to each group separately

- or, taking an average of the $S_i$, by

$$D^2(i) = (u - \bar{x}_i)' W^{-1} (u - \bar{x}_i)$$



- These measures of distance are termed ***generalized distance*** or **Mahalanobis distance**
- Historical note:

**Mahalanobis**:– Indian Statistician, key figure in Indian Independence movement, with Ghandi & Nehru
founder of Indian Statistical Institute
main work from ~1935 – 1960



- Instead of

$$D^2(i) = (u - \bar{x}_i)' W^{-1} (u - \bar{x}_i)$$

better to use

$$D^2(i) = (u - \bar{x}_i)' A_t A_t' (u - \bar{x}_i) = [A_t'(u - \bar{x}_i)]' [A_t'(u - \bar{x}_i)]$$

where $A_t$ is matrix of first t crimcoords
(i.e. project data into discriminant space & use Euclidean distance).

Not very different since $A_t$ crimcoords are eigenvectors of $W^{-1}B$



## Implementation in R

**To plot supplementary data in discriminant space (i.e. on crimcoords)**

```
> library(MASS)
> attach(irisnf)
> samp<- c(sample(1:50, 25), sample(51:100, 25),
  sample(101:150, 25))
> irisnftraining <- irisnf[samp]
> irisnftest<- irisnf[-samp]
> detach(irisnf)
> attach(irisnftraining)
> iristrain.lda<-
    lda(Variety~Sepal.l+Sepal.w+Petal.l+Petal.w)
> plot(iristrain.lda)
> points(predict(iristrain.lda,
  irisnftest[,-5])$x,pch=19)
```

**Code for selecting a [random] half of iris data, calculating the crimcoords and then plotting the other half as supplementary points on the crimcoords**



## Implementation in R

**To plot supplementary data in discriminant space (i.e. on crimcoords)**

```
> library(MASS)
```

• Attach library containing lda(.) function

```
> attach(irisnf)
```

• Attach data set

```
> samp<- c(sample(1:50, 25), sample(51:100,
25), sample(101:150, 25))
```

• create a vector **samp** which has 25 random numbers between 1 and 50, 25 in range 51 to 100 & 25 in 101 to150, i.e. a random half of numbers 1 to 150

```
> irisnftraining <- irisnf[samp,]
```

• take a random half of the data, ensuring half of each variety

```
> irisnftest<- irisnf[-samp,]
```

• take the other half of the data





```
> irisnftraining <- irisnf[samp,]
> irisnftest<- irisnf[-samp,]
```

- **irisnftraining** will be used for calculating the lda & then want to classify **irisnftest** by plotting these on the crimcoords

```
> detach(irisnf)
> attach(irisnftraining)
```

- switch current data set to the test data **irisnftraining**

```
> irisnftrain.lda<-
lda(Variety~Sepal.l+Sepal.w+Petal.l+Petal.w)
```

- calculate the lda using **irisnftraining**

```
> plot(irisnftrain.lda)
```

- plot **irisnftraining** on the crimcoords

```
> points(predict(irisnftrain.lda,
+ irisnftest[,-5])$x,pch=19)
```

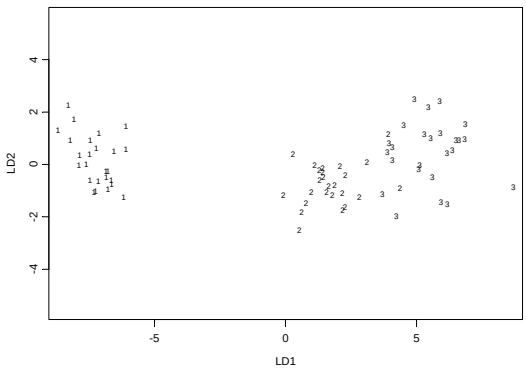
- add in the test data **irisnftest** on the same crimcoords



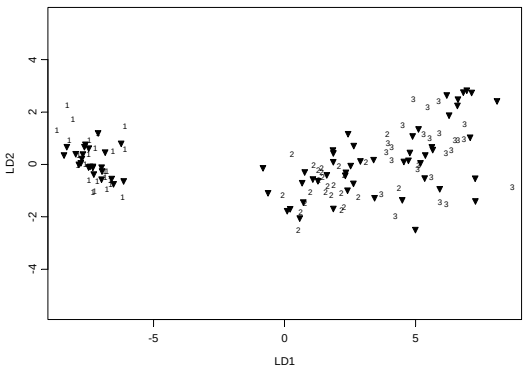
```
> points(predict(irismftrain.lda,
+ irisnftest[,-5])$x,pch=19)
WARNING: Point out of bounds: x = -9.346876,
y = 1.295536
```

- add in the test data **irisnftest** on the same crimcoords
- note the use of **points()** to add things to current plot & that one of these was outside the plotting area
  - (could have set up the plotting area from **irisnf** with **type="n"** to get the axes etc)
- note removal of the 5th column from **irisnftest** which has the varieties
- note use of **$x** to get the coordinates from the predicted data
- note use of **pch** to get a particular symbol










## Summary and Conclusions

- Crimcoords highlight differences between ***known*** groups of observations
  - strictly speaking, displays distorting data slightly (axes not orthogonal but conventionally drawn so)
- The first Crimcoord is also known as Fisher's Linear Discriminant Function
- PCA analogy gives interpretations of factor loadings and use of scree plots



- Plots on crimcoords gives informal classification of new cases
- Other classification methods use minimum Mahalanobis distance
- See Appendix 1 for further illustrations of linear and quadratic discriminant analysis
- Classification Trees and Neural Networks are related techniques of ***supervised learning***
  (see Appendices 7 & 8)
  and *Logistic Regression*
  (see other courses)