

Lecture 4

Interactive graphics for data analysis

Multidimensional scaling

Investigate several dimensions

- A process depends on the factors $X_1 \dots X_p$ where $p > 3$
- It is impossible to produce p-dimensional plot (recall: it was not even possible to understand static 3D-scatterplot)
- **Conclusion:** More advanced methods are needed
- Investigating:
 - Important factors
 - Connection between factors
 - Outliers
 - Clusters

Tools

- There are special tools to create **interactive** and **dynamic** plots.

Commercial:

- **Spotfire** – Many static and interactive visualization tools
- **Tableau**– Many static and interactive visualization tools
- **SAS JMP®** diagrams with rotations, linked diagrams, brushing, good user interface

Free

- **GGobi**: diagrams with rotations, linked diagrams, brushing, (badly documented, a few plot types)
- **Manet**: focus on the analysis of missing data, qualitative data
- **Shiny**: some interaction tools, requires programming skills.



GGobi

Link

<http://www.ggobi.org/>

Documentation:

<http://www.ggobi.org/docs/>

Principles:

<http://www.ggobi.org/book/>

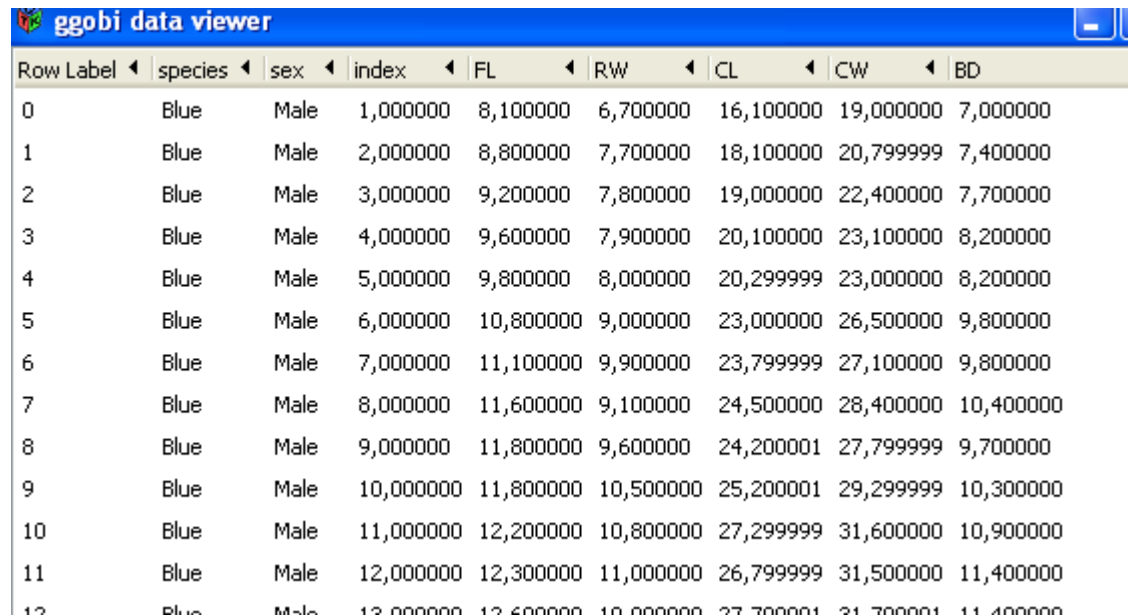
Interactive and Dynamic Graphics for Data Analysis: With Examples
Using R and GGobi. Dianne Cook and Deborah F. Swayne

Example 1

Australian crabs

Variables:

- Qualitative:
 - Species
 - Sex
- Quantitative:
 - Frontal lobe
 - Rear width
 - Carapace length
 - Carapace width
 - Body depth



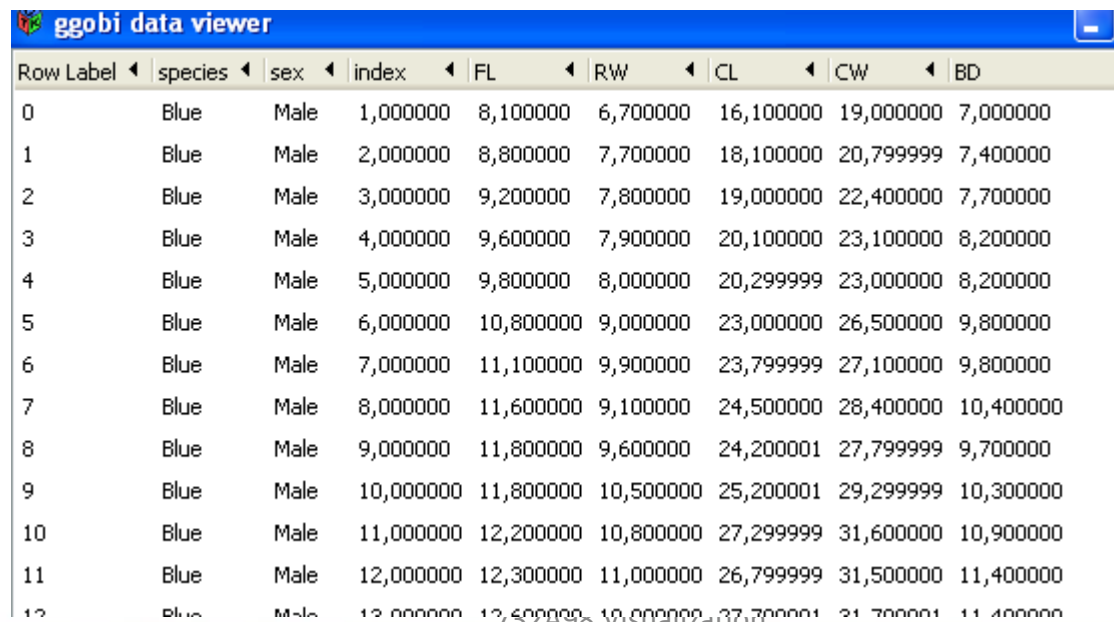
The image shows a screenshot of a software window titled "ggobi data viewer". The window displays a table with 10 columns: "Row Label", "species", "sex", "index", "FL", "RW", "CL", "CW", and "BD". The table contains 12 rows of data, all representing "Blue" crabs of "Male" sex. The numerical values for the quantitative variables (index, FL, RW, CL, CW, BD) increase incrementally from row 0 to row 12.

Row Label	species	sex	index	FL	RW	CL	CW	BD
0	Blue	Male	1,000000	8,100000	6,700000	16,100000	19,000000	7,000000
1	Blue	Male	2,000000	8,800000	7,700000	18,100000	20,799999	7,400000
2	Blue	Male	3,000000	9,200000	7,800000	19,000000	22,400000	7,700000
3	Blue	Male	4,000000	9,600000	7,900000	20,100000	23,100000	8,200000
4	Blue	Male	5,000000	9,800000	8,000000	20,299999	23,000000	8,200000
5	Blue	Male	6,000000	10,800000	9,000000	23,000000	26,500000	9,800000
6	Blue	Male	7,000000	11,100000	9,900000	23,799999	27,100000	9,800000
7	Blue	Male	8,000000	11,600000	9,100000	24,500000	28,400000	10,400000
8	Blue	Male	9,000000	11,800000	9,600000	24,200001	27,799999	9,700000
9	Blue	Male	10,000000	11,800000	10,500000	25,200001	29,299999	10,300000
10	Blue	Male	11,000000	12,200000	10,800000	27,299999	31,600000	10,900000
11	Blue	Male	12,000000	12,300000	11,000000	26,799999	31,500000	11,400000
12	Blue	Male	13,000000	12,600000	10,000000	27,700001	31,700001	11,400000

Introduction to GGobi

Open data:

- Format is "csv" or xml
- File → Open ...
- Tools → Data viewer

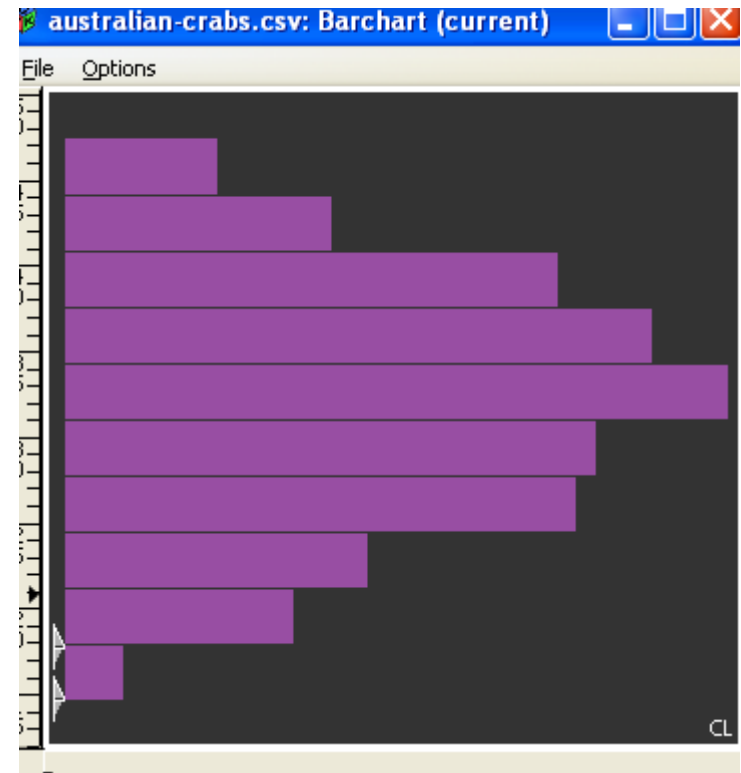


The screenshot shows the 'ggobi data viewer' window. It contains a table with 10 columns: Row Label, species, sex, index, FL, RW, CL, CW, and BD. The data is organized into 12 rows, with the last row partially cut off. The species are all 'Blue' and the sex is 'Male' for all entries. The index values range from 1,000,000 to 12,000,000. The other columns (FL, RW, CL, CW, BD) contain numerical values that generally increase with the index.

Row Label	species	sex	index	FL	RW	CL	CW	BD
0	Blue	Male	1,000000	8,100000	6,700000	16,100000	19,000000	7,000000
1	Blue	Male	2,000000	8,800000	7,700000	18,100000	20,799999	7,400000
2	Blue	Male	3,000000	9,200000	7,800000	19,000000	22,400000	7,700000
3	Blue	Male	4,000000	9,600000	7,900000	20,100000	23,100000	8,200000
4	Blue	Male	5,000000	9,800000	8,000000	20,299999	23,000000	8,200000
5	Blue	Male	6,000000	10,800000	9,000000	23,000000	26,500000	9,800000
6	Blue	Male	7,000000	11,100000	9,900000	23,799999	27,100000	9,800000
7	Blue	Male	8,000000	11,600000	9,100000	24,500000	28,400000	10,400000
8	Blue	Male	9,000000	11,800000	9,600000	24,200001	27,799999	9,700000
9	Blue	Male	10,000000	11,800000	10,500000	25,200001	29,299999	10,300000
10	Blue	Male	11,000000	12,200000	10,800000	27,299999	31,600000	10,900000
11	Blue	Male	12,000000	12,300000	11,000000	26,799999	31,500000	11,400000
12	Blue	Male	13,000000	12,600000	10,500000	27,700001	31,700001	11,400000

Available diagrams

- Histogram:
 - Display -> New barchart
 - Choose variable
 - Vary bin width directly at the plot

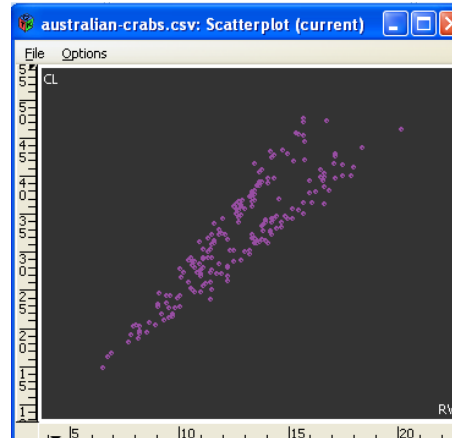
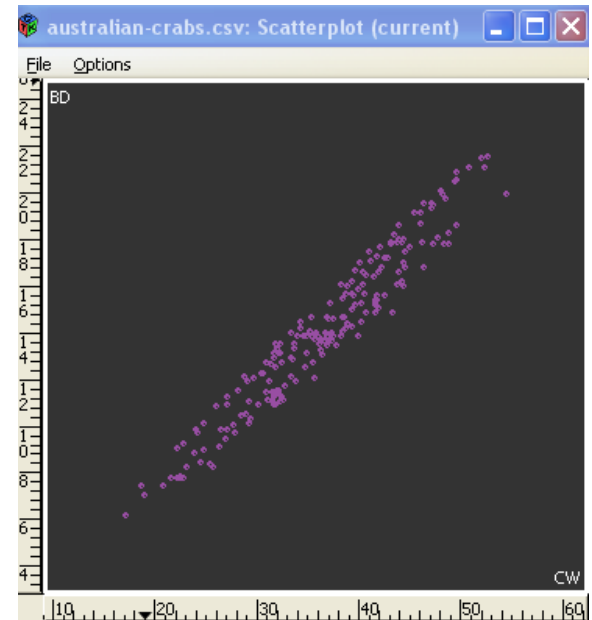


Available diagrams

- Scatterplot
- Choose X and Y
- ..or see all by using "Cycle"

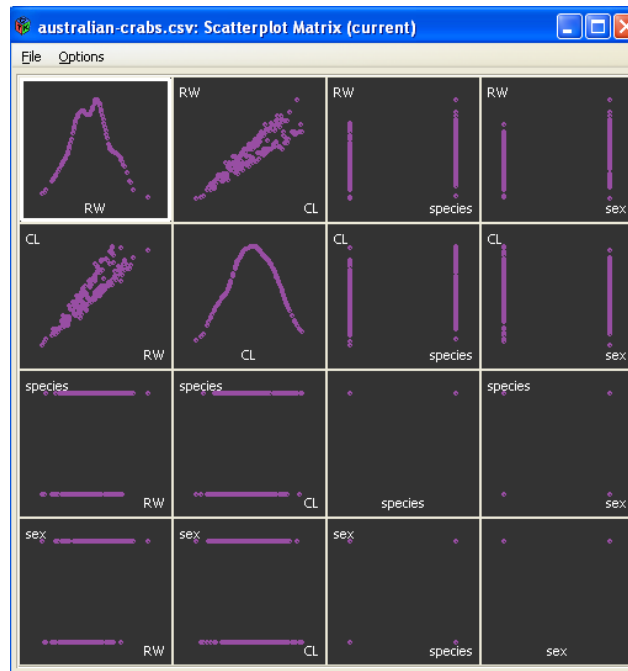
Interesting: CW vs RW

- Two clusters are seen



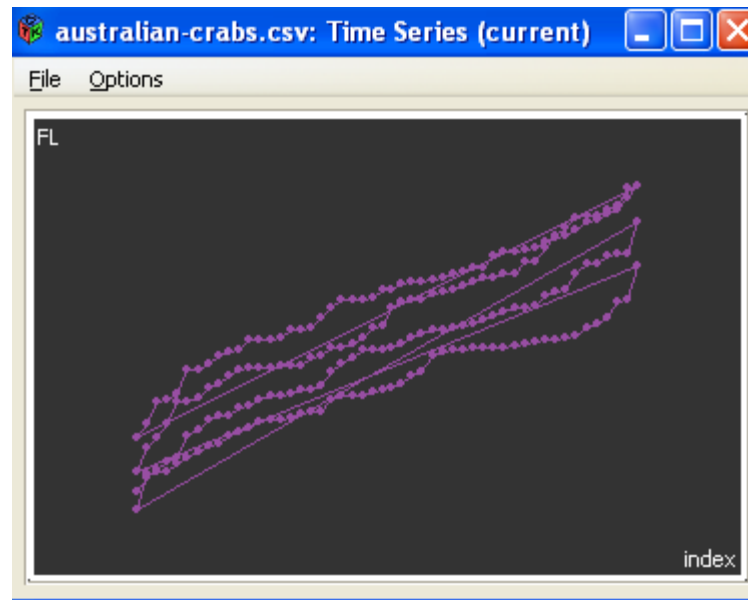
Available diagrams

- Alternative: scatterplot matrix
 - Display → New scatterplot matrix
 - Choose variables that should be used



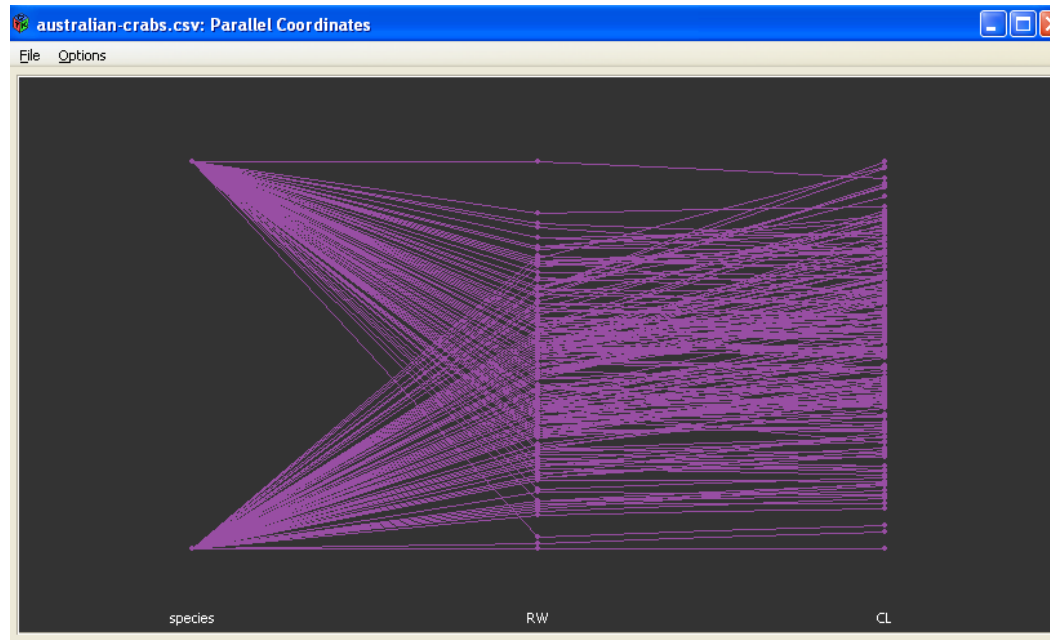
Available diagrams

- Time series plot
 - Choose time axis X
 - Choose one or more Y



Available diagrams

- Benefit: plot is interactive
 - each trace line can be identified
 - a group of trace lines can be selected and brushed



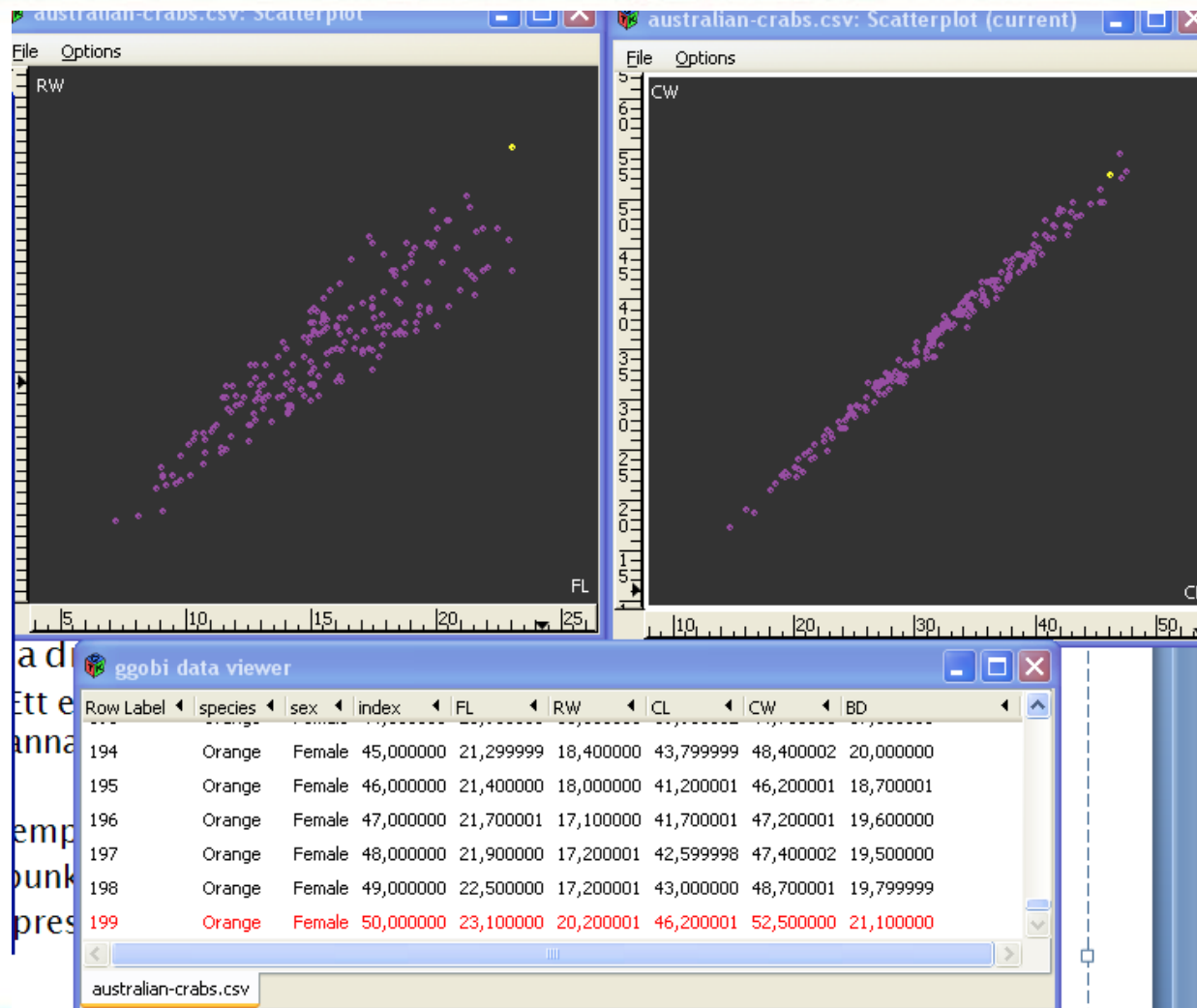
Linked plots & Brushing

- All plots created in one Ggobi sessions are linked, i.e.
 - Each element (observation) in one plot corresponds to one or more elements in the other plot

Example: 2 scatterplots FR vs RW and CL vs CW

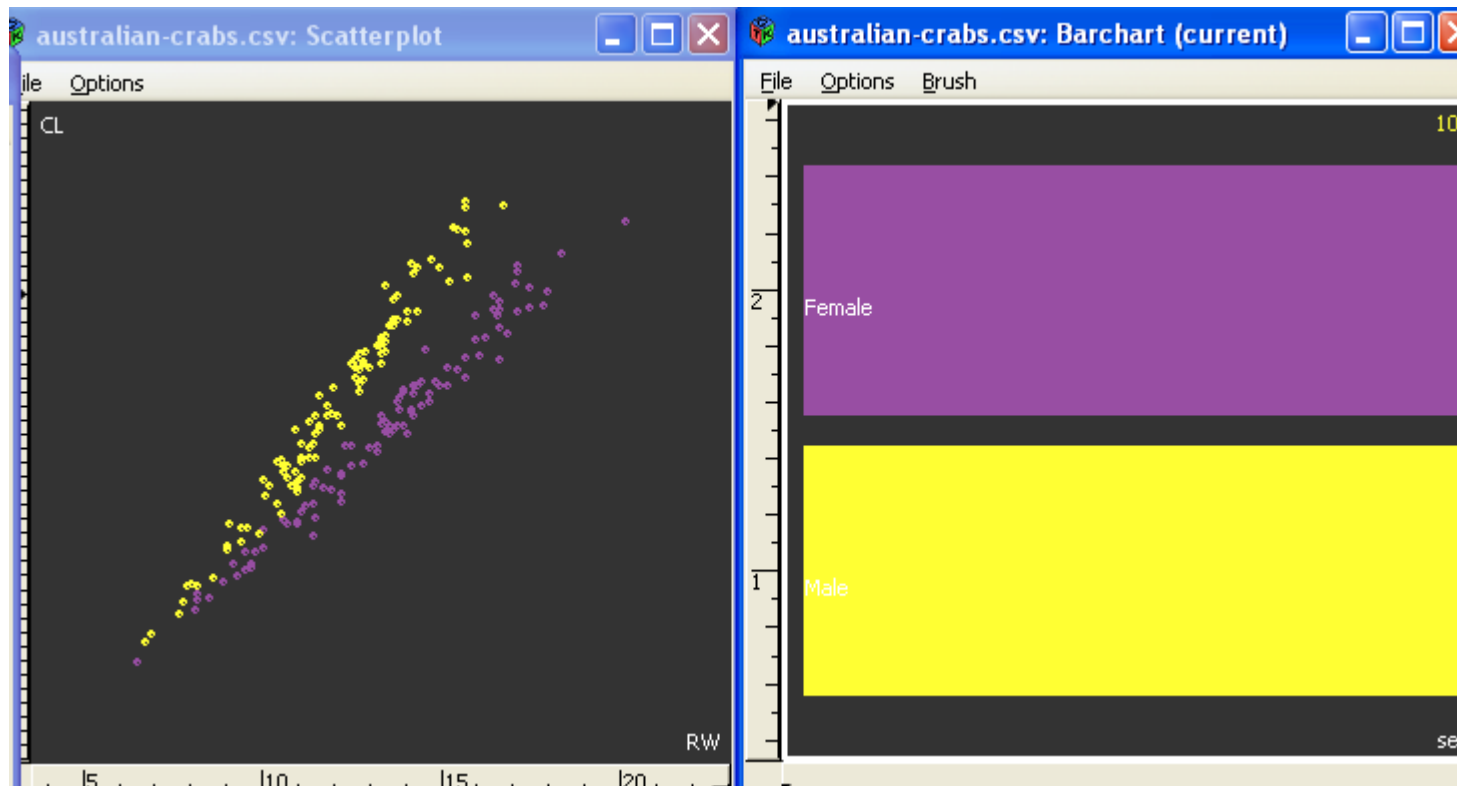
1 one observation (FR,RW) corresponds 1 observation (CL,CW)

Linked plots & Brushing



Linked plots & Brushing

Example: 1 scatterplot **FR** vs **RW** and 1 histogram **sex**
1 bar **sex** corresponds several observations (CL,RW) –



Brushing

- **Brushing** implies that the process of the coloring of items in one plot leads to the automatic coloring of the corresponding items in the linked plots

Notions:

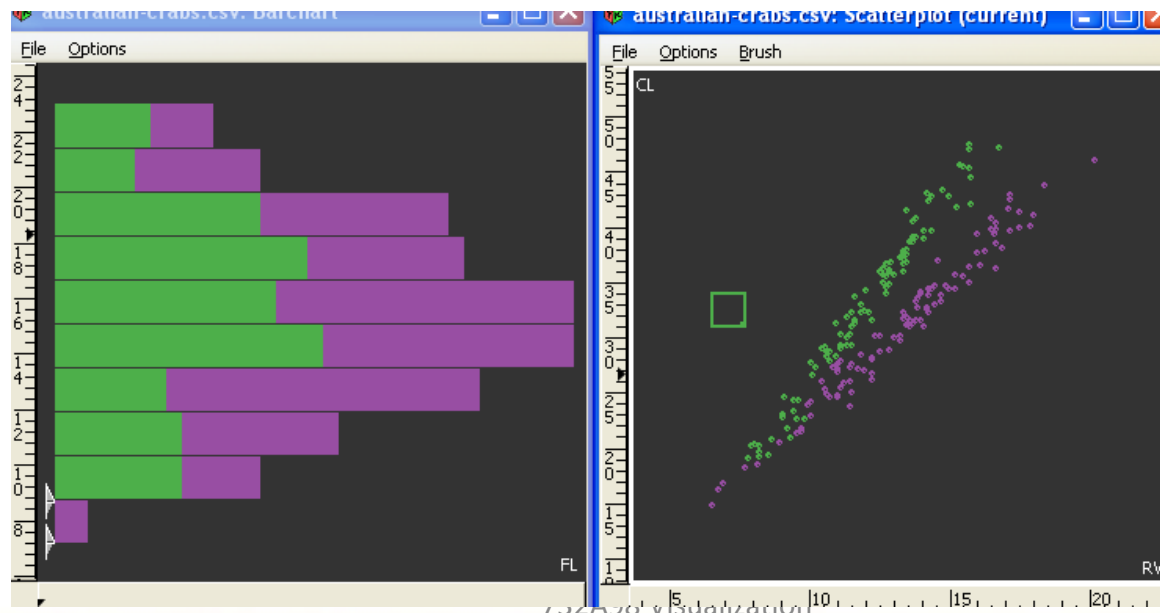
1. Active window (white borders) all operations here
2. Brush
3. Persistent and transient brushing (to fix clusters)

GGobi: Interaction -> brush

Brushing

Example: 1 scatterplot **CL** vs **RW** and 1 histogram (**different vars**)

- Choose green brush, normal size
- Color one cluster and see how histograms change
- See where extreme cases of FL are located in (CL, RW)
- Conclusions?

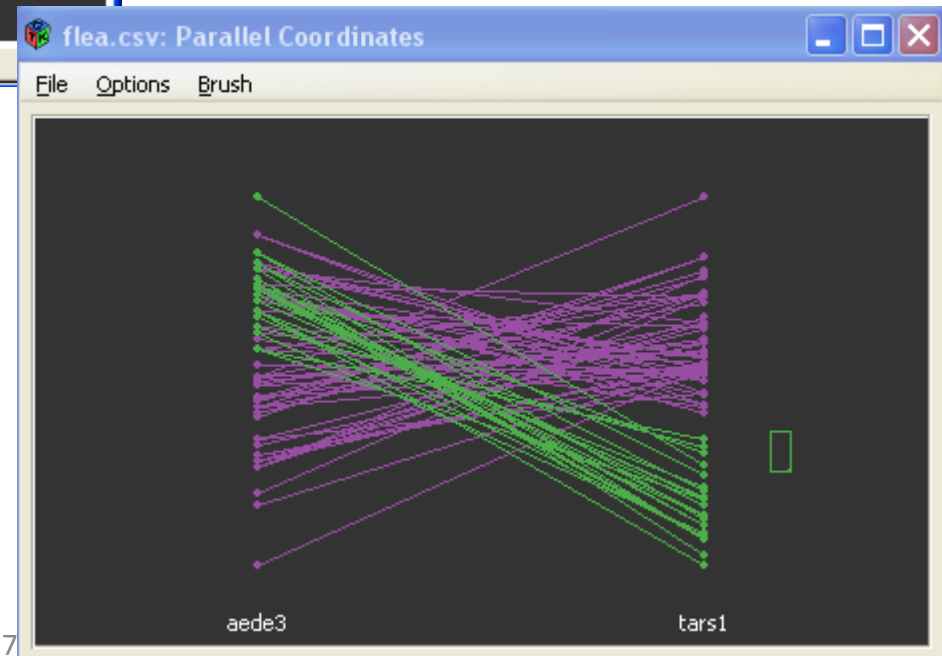
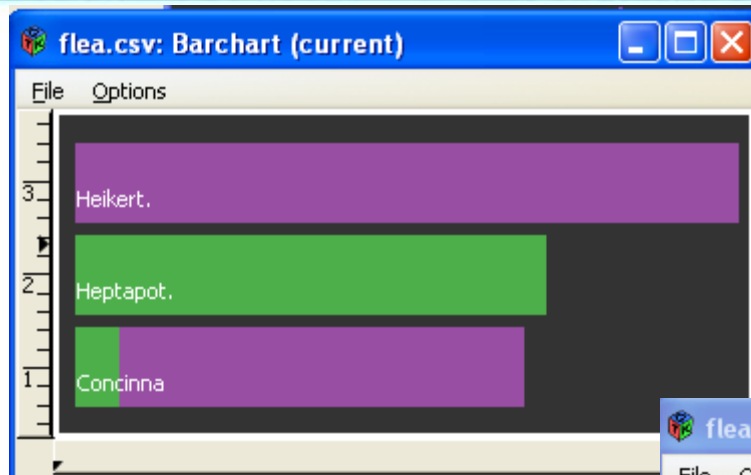


The top of the slide features a blue background with a pattern of binary code (0s and 1s). A magnifying glass is positioned over the text 'Example 2', focusing on it.

Example 2

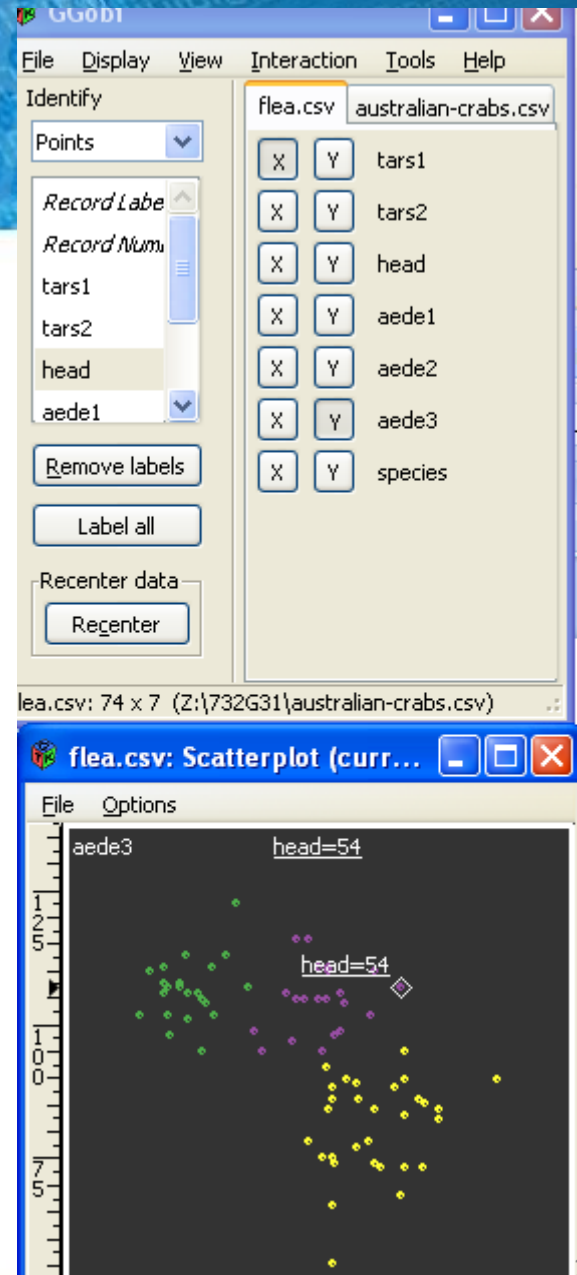
- Flea beetles (flea.csv)
- 6 variables
- Aim is to identify different arts based on beetle measurements

Example 2: one cluster detected



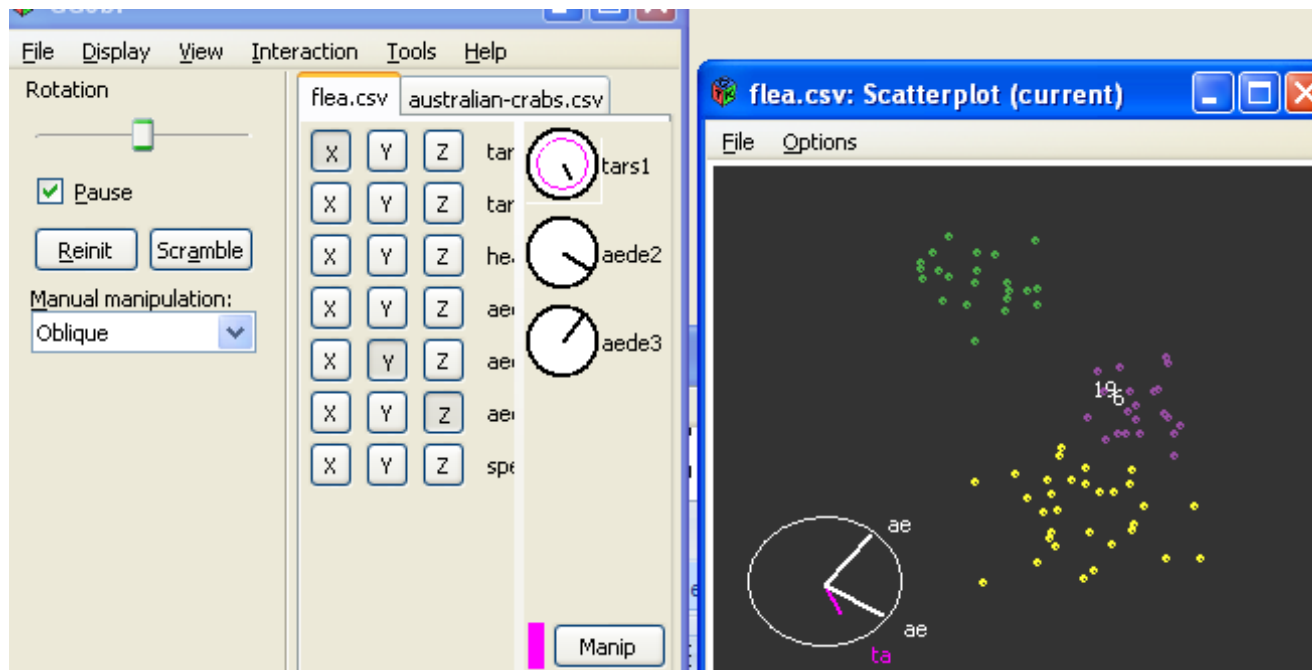
Identification

- Aim is to identify the item on the plot, for ex. ID or the value of an outlier
- Interaction -> Identify
- What can be seen:
 - Record Number or Record Label – ID in the database
 - Choose a parameter to see its value
- Click on the observation to make the label permanent
- Identification is done in the linked plots too



3D-rotation

- For a scatterplot, choose View→ Rotation
- Choose X, Y,Z
- Click "Scramble" to change projection randomly
- Variable circle shows angle between axis and projection plane

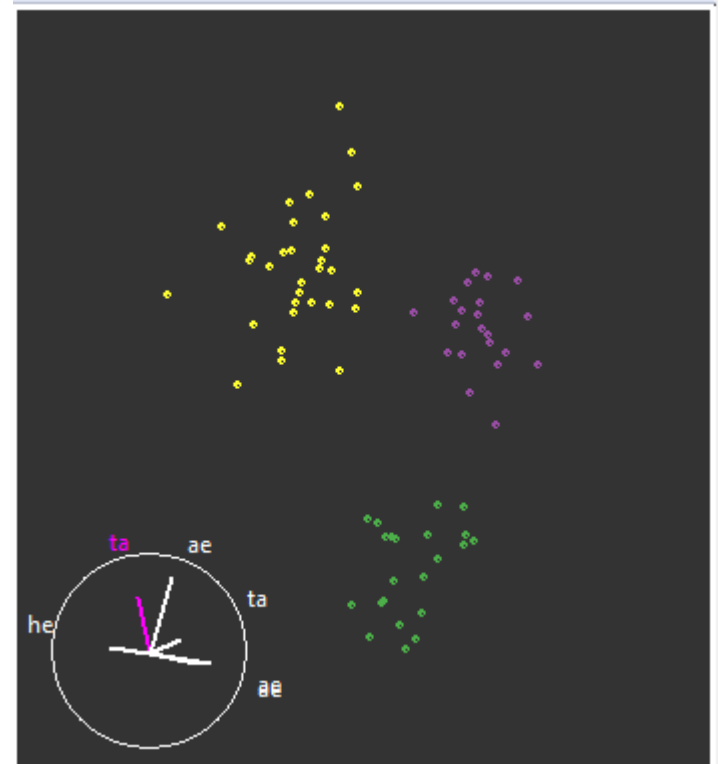


2D tour

- View → 2D tour
- Data is projected into two dimensions
- Sometimes, clusters are seen

Example:

1. Use Flea.csv with colored observations
2. Try to see clusters



Distance between objects

- Meaning of "two objects are close"?
 - Measure of proximity (ex: quantitative vars, Euclidian distance)
- Similarity measure s_{rs} (=1 if same object, <1 otherwise)
 - Ex: correlation
- Dissimilarity measure δ_{rs} (=0 if same object, >0 otherwise)
 - Ex: euclidian distance

Problem of constructing the measures of proximity:

- What if the variable is qualitative?
- What if the object is a text document?

Multidimensional scaling

- Given n object with known matrix of similarities of dissimilarities. Each object i is characterized by p -dimensional vector X_i

The aim:

Present these objects in lower dimensions ($p'=2$ or 3) such that the distance between the new points would reflect the matrix of similarities (or dissimilarities)

- See neighbour observations
- See clusters and outliers
- Have a "map" of your data

Two types:

- Metric MDS
- Non-metric MDS

Multidimensional scaling

Metric MDS (algorithm is not discussed here)

Searching for points $\chi_1 \dots \chi_n$, such that:

If $\delta_{rs} \leq \delta_{qt} \rightarrow d_{rs} \leq d_{qt}$ (ranks are preserved)

Non-metric MDS

Given n objects X_1, \dots, X_n with known matrix of similarities $||\delta_{rs}||$ of dissimilarities.

Step 1: For some configuration $\chi_1 \dots \chi_n$ (in lower dimension) with matrix $||d_{rs}||$, define primary monotonic regression $\chi'_1 \dots \chi'_n$ with matrix $||d'_{rs}||$ such that $\chi'_1 \dots \chi'_n$ as close as possible to $\chi_1 \dots \chi_n$ and if $\delta_{rs} \leq \delta_{qt} \rightarrow d'_{rs} \leq d'_{qt}$

(*Meaning:* MR gives points close to the configuration $\chi_1 \dots \chi_n$, and reflecting original dissimilarities)

Multidimensional scaling

Non-metric MDS

Step 2: For each configuration $\chi_1 \dots \chi_n$ with matrix $||d_{rs}||$, define stress:

$$S = \sqrt{\frac{\sum_{r,s} (d_{rs} - d'_{rs})^2}{\sum_{r,s} d_{rs}^2}}$$

Step 3. Consider $S=S(\chi_1 \dots \chi_n)$ and minimize it using optimization methods (e.g. steepest descend) \rightarrow Find best configuration

Metric multidimensional scaling

1. Compute dissimilarity (distance) matrix
 2. Use function **cmdscale(d,k)**
 - d is distance matrix
 - k is number of desired dimensions
- Music data
 - Artist (abba. Beatles. Wiwaldi, Mozart, Beethoven, Enya)
 - Type (rock, classical, new wave)
 - lvar, lave, lmax, lfener, lfreq – parameters of the music signal

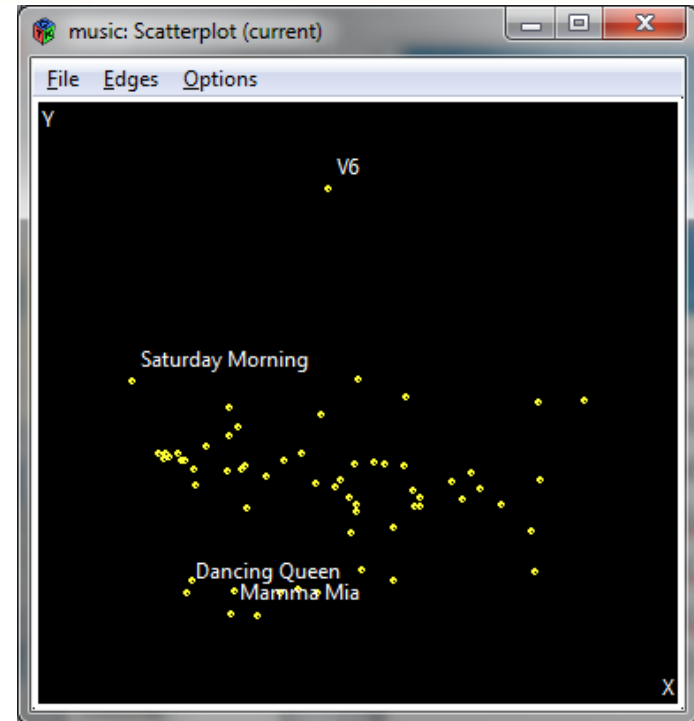
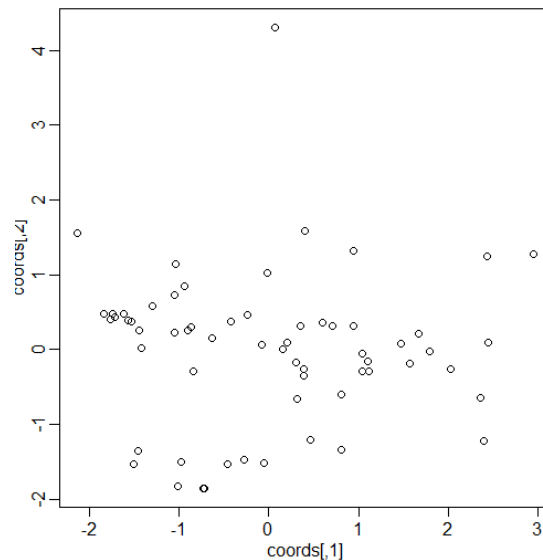
Metric multidimensional scaling

- **Codes:**

```
setwd("Z:/732A39/2014/Lecture 4")  
music = read.csv("music-sub.csv", row.names=1)  
music.numeric= scale(music[,4:7])  
d=dist(music.numeric)  
coords=cmdscale(d,2)  
plot(coords)
```

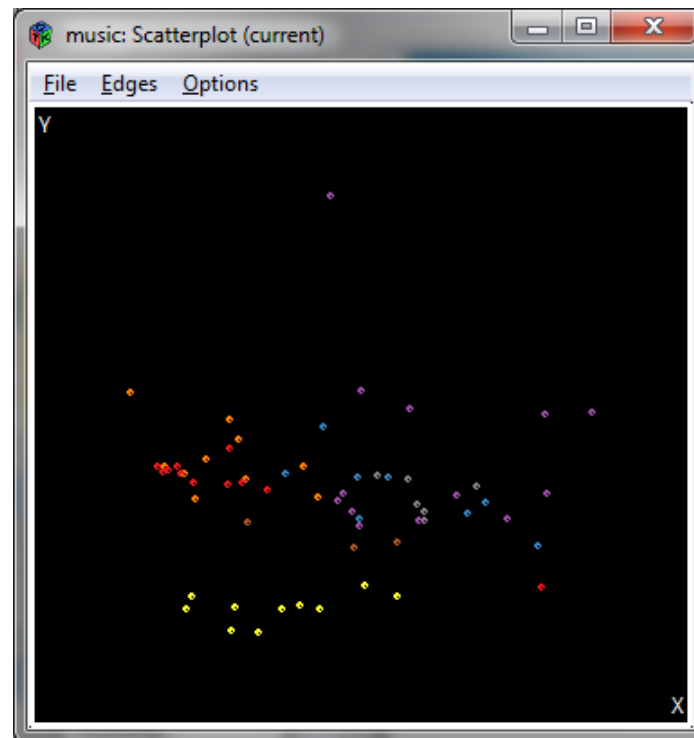
#or?

```
library(rggobi)  
ggobi(coords)
```



Metric multidimensional scaling

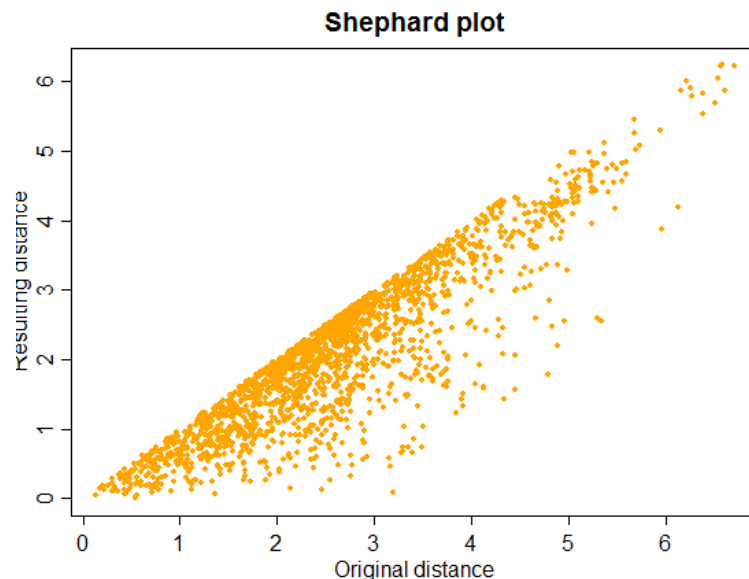
- Brushing by artist:
 - We can see yellow cluster=Abba



Shepard plot

- Y =distance between resulting points
- X : distance between original points
- Shows how good MDS reflects original points
 - Best: a monotonic line (seldom in practice)

```
#Shepard plot  
plot(d,dist(coords))
```

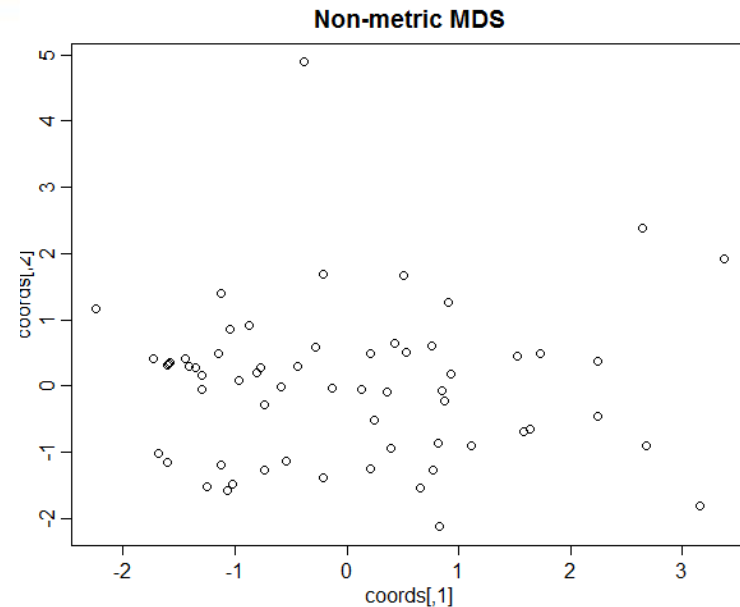
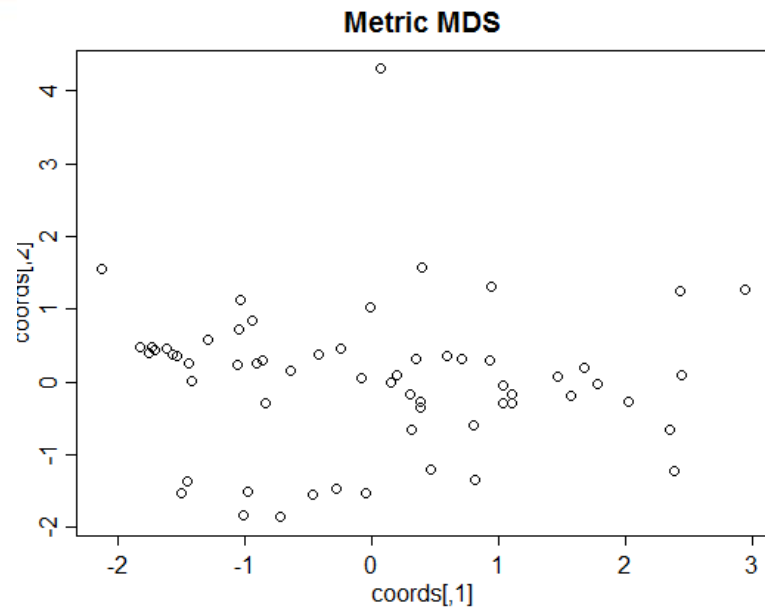


Nonmetric MDS

1. Compute dissimilarity (distance) matrix
2. Use function **isoMDS(d,y,k,p)** in library MASS
 - d is distance matrix
 - y: starting points (not necessary, but the result is sensitive to this choice!)
 - k is number of desired dimensions
 - p – power of the Minkowski distance
 - p=2 - Euclidian

```
library(MASS)
music = read.csv("music-sub.csv",
row.names=1)
music.numeric= scale(music[,4:7])
d=dist(music.numeric)
res=isoMDS(d,k=2)
coords=res$points
plot(coords)
```

Nonmetric MDS



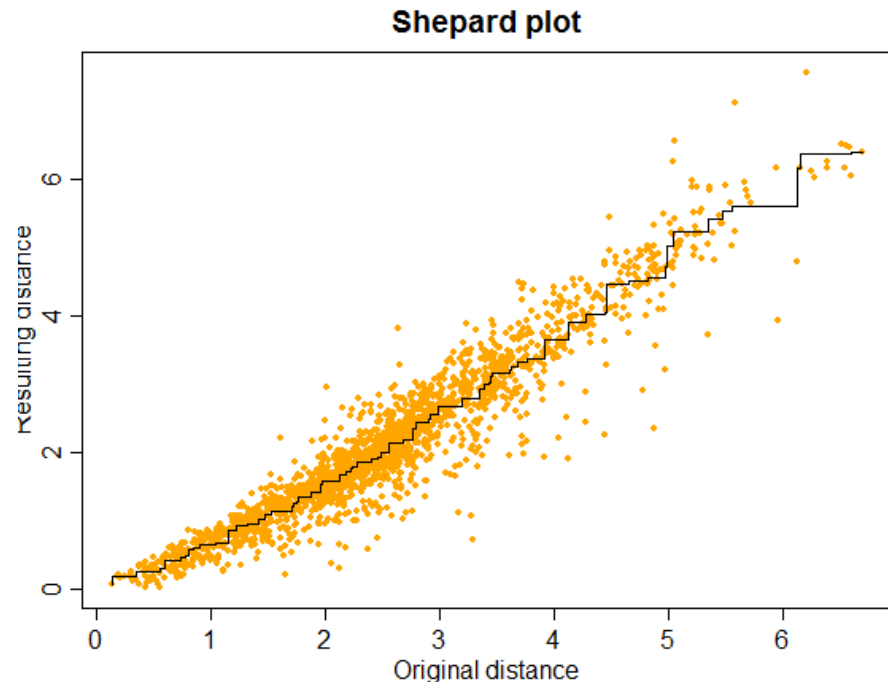
```
> res=isomDS(d,k=2)
initial value 21.040048
iter 5 value 14.342233
final value 14.255305
converged
```

← Stress values

Shepard plot

- In non-metric MDS, we also add fitted values of the monotonic configurations ($\chi'_1 \dots \chi'_n$)

```
sh <- Shepard(d, coords)
plot(d, dist(coords))
lines(sh$x, sh$yf, type = "S")
```



To read at home

- Corresponding parts in Ggobi manual

<http://www.ggobi.org/docs/manual.pdf>

- Paper on MDS (see LISAM)