Statistics 120
Clusters and Surfaces

High-Dimensional Data

- A typical statistical data set can be laid out in a *matrix* with rows corresponding to cases and columns to variables.
- If there are n cases and p variables:

$$\begin{array}{cccc} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{array}$$

Data Coordinates

- The i-th row of the data matrix can be regarded as specifying a point in a p-dimensional space.

$$(x_{i1},\ x_{i2},\ \ldots\ ,\ x_{ip})$$

- For $p = 1$, 2 and 3 it is possible to represent these coordinates directly.
- For $p > 3$ our intuition fails us and we must consider indirect methods.

The Iris Data

- This set of data was collected by a botanist - Edgar Anderson.
- It gives the widths and lengths of the petals and sepals of three species of Iris:
  - *Iris Setosa*
  - *Iris Versicola*
  - *Iris Virginica*
- The dataset is often used to test statistical techniques which attempt to distinguish different groupings on the basis of measurements.

Iris Flowers

Iris Setosa

Iris Versicolor

Iris Virginica

Unfortunately, the data set doesn’t contain the most important information about the Iris flowers.

The Iris Data

Sepal Length	Sepal Width	Petal Length	Petal Width
5.1	3.5	1.4	0.2
4.9	3.0	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5.0	3.6	1.4	0.2
5.4	3.9	1.7	0.4
4.6	3.4	1.4	0.3
5.0	3.4	1.5	0.2
4.4	2.9	1.4	0.2
4.9	3.1	1.5	0.1
⋮	⋮	⋮	⋮

Scatterplot Matrices (Draughtsman's Displays)

- A simple way to examine high dimensional datasets is to plot all possible pairs of variables.
- There are $p \times (p-1)$ scatter plots to be viewed.
  - There are p choices for the x variable.
  - For each x variable there are $p-1$ possible choices for the y variable.
- One way to display the plots is to lay them out a $p \times p$ matrix.
- This kind of display is called a *scatterplot matrix* or a *draughtsman's display*.

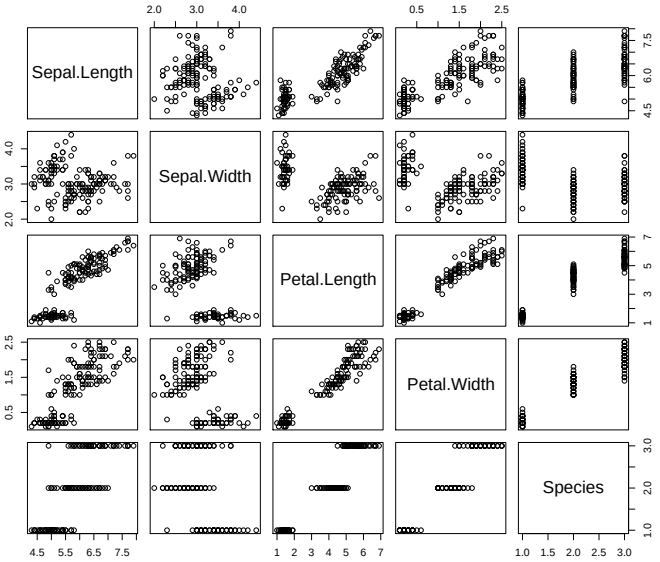
Scatterplot Matrices in R

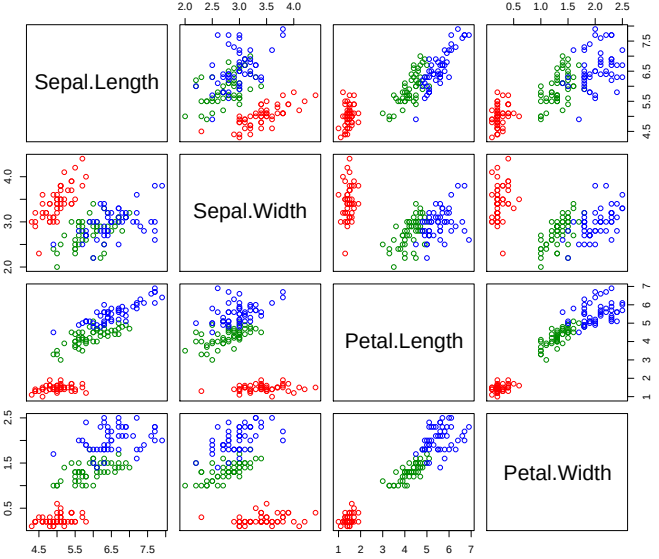
- The R function `pairs` produces a scatterplot matrix.

```
> data(iris)
> pairs(iris)
```

- The function allows a degree of customisation – plotting symbol and default colour can be easily changed.

```
> cols = rep(c("red", "green4", "blue"),
      c(50, 50, 50))
> pairs(iris[, 1:4], col = cols)
```


Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species


Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
2.0 2.5 3.0 3.5 4.0
0.5 1.0 1.5 2.0 2.5
4.5 5.5 6.5 7.5
1 2 3 4 5 6 7

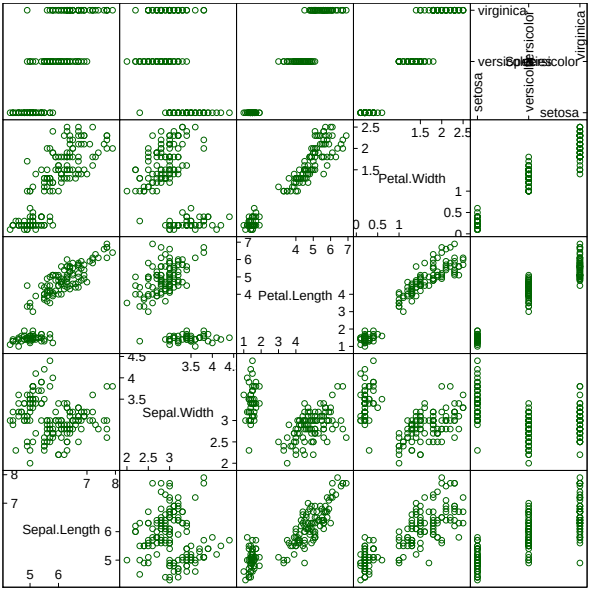
Scatterplot Matrices in Trellis

- It is also possible to produce scatterplot matrices using the Trellis function `splom`.

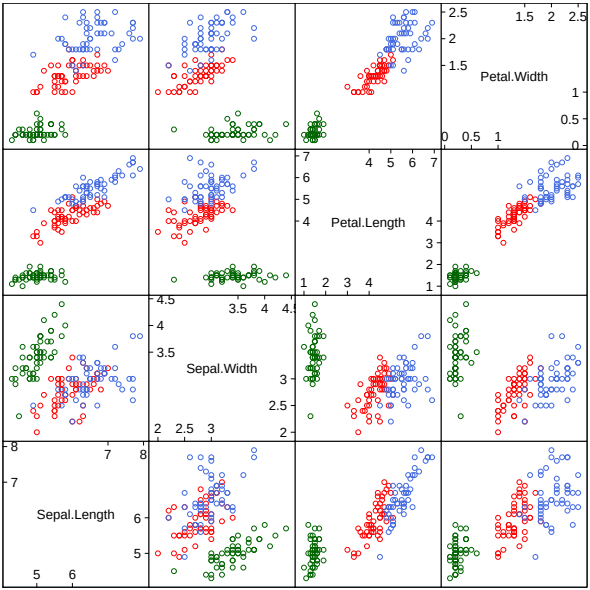
```
> library(lattice)
> lset(col.whitebg())
> splom(iris)
```

- Customising `splom` is possible, but a little more complex than the customising `pairs`.

```
> splom(iris[, 1:4], group = iris$Species,
      pch = 1, panel = panel.superpose)
```



Scatter Plot Matrix

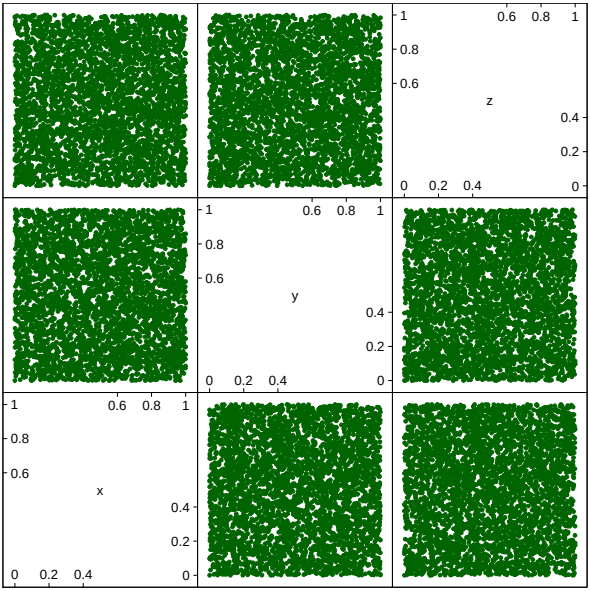


Scatter Plot Matrix

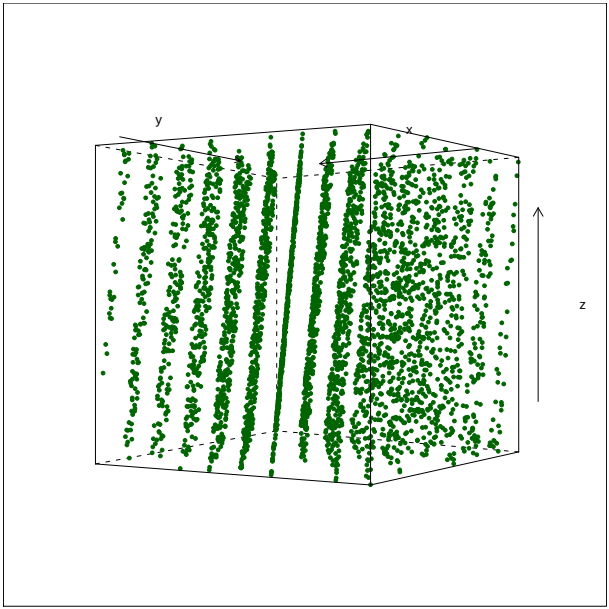
Limitations of Scatterplot Matrices

- Scatterplot matrices can give a good overall view of a set of data values, but they also be misleading.
- This is because they only show a very limited view of the data. To illustrate the problem, we will look at the "randu" dataset.
- This data set consists of consecutive triples produced by the randu random number generator.

```
> data(randu)
> splom(randu)
> cloud(y ~ x * z, data=randu,
        screen = list(y = 147),
        perspective = FALSE)
```



Scatter Plot Matrix


y
x
z

Comments

- The consecutive triples produced by randu are constrained to lie on a series of parallel planes which cut through the unit cube.
- The paper which pointed this fact out was titled “The Random Numbers Fall Mainly on the Planes.”
- The planes are not aligned with the sides of the unit cube and so do not show up in any of the panels of the scatter plot.
- This problem can be even worse in higher dimensions.

Clustering

- One of the ways we seek to make sense of the world around us is by grouping the things we see about us into classes of similar objects.
- If the objects in a group are sufficiently similar and sufficiently distinct from other objects we may give them a common name — person, dog, chair, etc.
- In a further step, we may begin to create theories about the relationships between groups.
- In statistics, forming groups of similar objects is known as *cluster analysis* or *clustering*.

Clustering and Graphics

- There are a number of graphical techniques which aim to help users establish the degree to which observations are similar or different.
- All these techniques work by encoding each observations as a symbol or *glyph*.
- The visual system is very good at letting us detect visual simularity.
- This can form the basis for informally clustering observations.

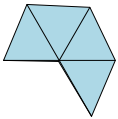
Example – United States Voting

Percentage of Republican Votes
in Presidential Elections in Six Southern States
in the Years 1932–1940 and 1960–1968.

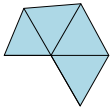
	1932	1936	1940	1960	1964	1968
Missouri	35	38	48	50	36	45
Maryland	36	37	41	46	35	42
Kentucky	40	40	42	54	36	44
Louisiana	7	11	14	29	57	23
Mississippi	4	3	4	25	87	14
South Carolina	2	1	4	49	59	39

Stars — A Simple Glyph

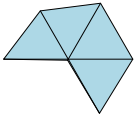
- One simple way of encoding the vote data is to draw a star with one arm for each voting year.
- The lengths of the arms will be proportional to the vote for the corresponding year.
- Each State will be encoded as a six-pointed star.



Missouri

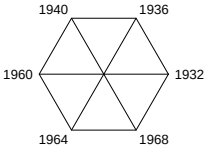


Maryland



Kentucky

Louisiana



Mississippi

South Carolina

Interpretation

- Clearly Missouri, Maryland and Kentucky exhibit very similar voting patterns.
- They can be regarded as forming a cluster.
- Louisiana, Mississippi and South Carolina are different from each other and the other cluster.
- Many other glyphs have been proposed.

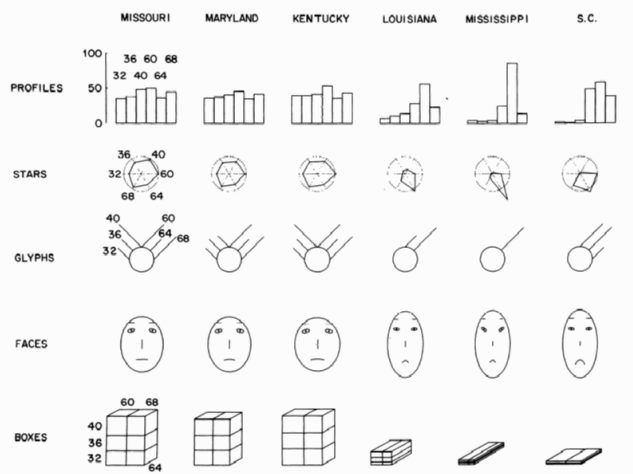


Figure 1. Profiles, Stars, Glyphs, Faces, and Boxes of Percentage of Republican Votes in Six Presidential Elections in Six Southern States. The Circles in the Stars Are Drawn at 50%. The Assignment of Variables to Facial Features in the Faces Is: 1932—Shape of Face; 1936—Length of Nose; 1940—Curvature of Mouth; 1960—Width of Mouth; 1964—Slant of Eyes; 1968—Length of Eyebrows

Critique

- Glyphs work well when there are just a few observations.
- With even moderate numbers of observations the ability of the brain to group the observations is overwhelmed.
- Little is known about how well our interpretation of the similarity of glyphs corresponds to the true similarity between the observations.
- In the case of faces, there are likely to be strong cultural and gender biases in an individuals groupings.

Examining Surfaces

- Surfaces are typically defined implicitly by an equation of the form:
$$f(x,y,z) = 0$$
- Only those points which satisfy the equation lie on the surface.
- The equation $x^2 + y^2 + z^2 - 1 =$ defines a sphere.
- The equation $ax + by + cz - 1 = 0$ defines a plane.

Explicit Formulation

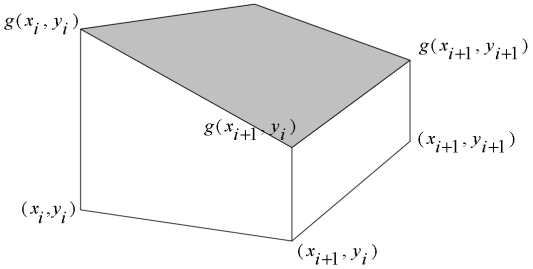
- Some surfaces can be defined *explicitly* by an equation of the form:
$$z = g(x,y)$$
- This is clearly a special case because if we define $f(x,y,z) = g(x,y) - z$ then
$$f(x,y,z) = 0$$
- Such surfaces occur often in scientific work and it is useful to have a way of drawing them.

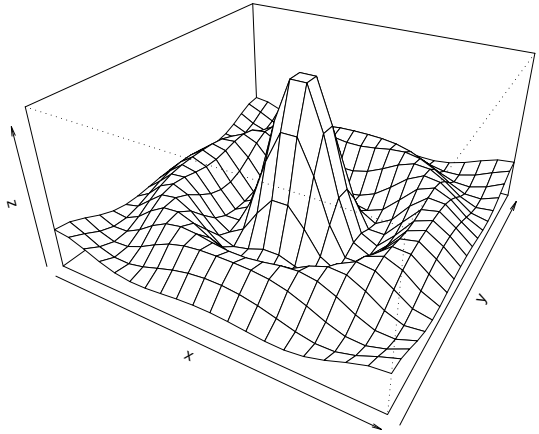
Drawing Surfaces

- To draw $z = g(x,y)$ over the region defined by $x \in [x_L, x_U]$ and $y \in [y_L, y_U]$.
- Partition the intervals $[x_L, x_U]$ and $[y_L, y_U]$.

$$x_L = x_0 < x_1 < \quad \ldots \quad < x_m = x_U$$
$$y_L = y_0 < y_1 < \quad \ldots \quad < y_n = y_U$$

- Approximate the surface by facets.


$g(x_i, y_i)$
$g(x_{i+1}, y_{i+1})$
$g(x_{i+1}, y_i)$
(x_{i+1}, y_{i+1})
(x_i, y_i)
(x_{i+1}, y_i)


z
x
y

Drawing A Surface in R

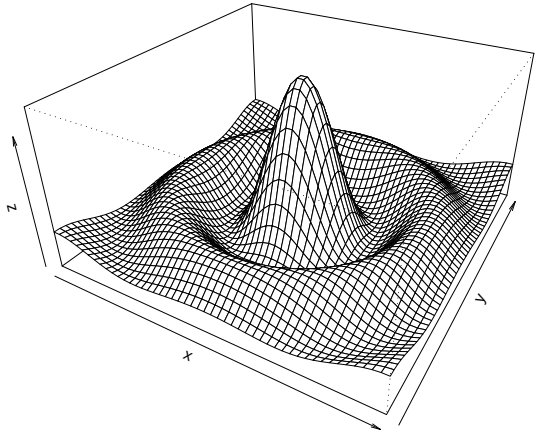
```
> f = function(x, y) {
    r = sqrt(x^2 + y^2)
    ifelse(r ==0, 1, 10 * sin(r)/r)
  }

> x = seq(-10, 10, length=20)
> y = x
> z = outer(x, y, f)

> persp(x, y, z, theta = 30, phi = 30,
        expand = 0.5)
```

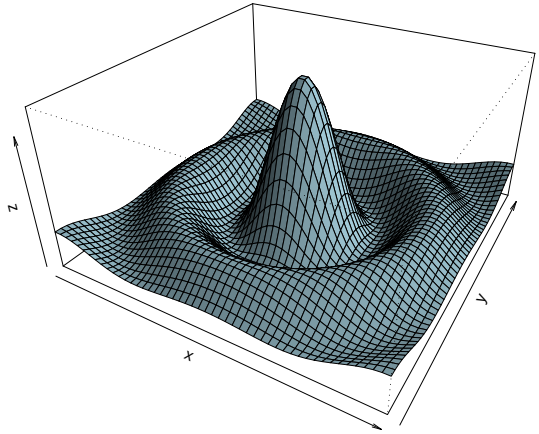
A Refined Grid

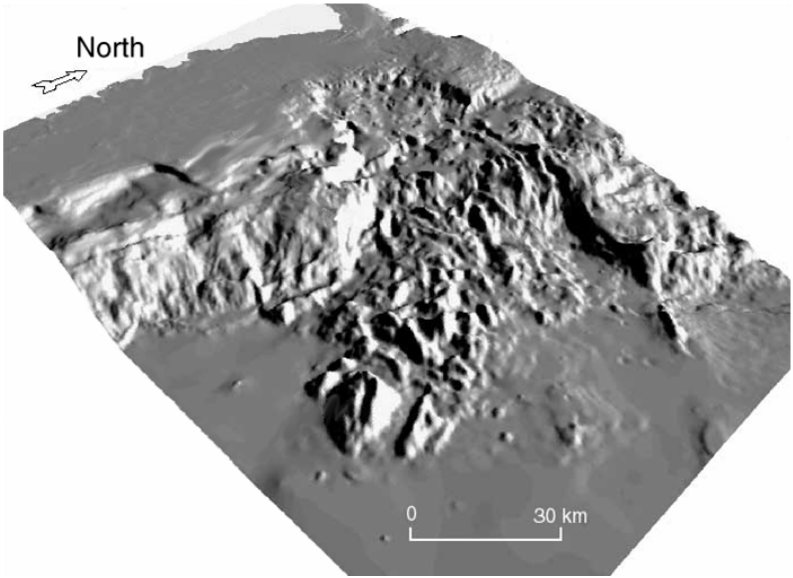
```
> x = seq(-10, 10, length=50)
> y = x
> z = outer(x, y, f)
> persp(x, y, z, theta = 30, phi = 30,
        expand = 0.5, col = "white")
```


z
x
y

Adding A Simple Lighting Model

```
> persp(x, y, z, theta = 30, phi = 30,
        expand = 0.5, col = "lightblue",
        ltheta = 120, shade = 0.5)
```


z
x
y


North
0
30 km