# Data Mining & Organization: Iris and other data sets

## Data understanding and visualization

Chapter 3, Introduction to Data Mining by Tan, Steinbach, Kumar

Donatella Merlini

Università di Firenze
Corso di Laurea Magistrale in Informatica
Curriculum Data Science

# The Iris data set

Iris Setosa

Iris Versicolor

Iris Virginica

- Collected by E. Anderson in 1935
- Contains measurements of four real-valued variables: sepal length, sepal widths, petal lengths and petal width of 150 iris flowers of types Iris Setosa, Iris Versicolor, Iris Virginica (50 each).
- The fifth attribute is the name of the flower type.

# Visualization

- Visualization is the conversion of data into a visual or tabular format so that the characteristics of the data and the relationships among data items or attributes can be analyzed or reported.
- Visualization of data is one of the most powerful and appealing techniques for data exploration.
  - Humans have a well developed ability to analyze large amounts of information that is presented visually.
  - Can detect general patterns and trends.
  - Can detect outliers and unusual patterns.

| slength | swidth | plength | pwidth | class |
|---|---|---|---|---|
| 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| . . . | | | | |
| 5.0 | 3.3 | 1.4 | 0.2 | Iris-setosa |
| 7.0 | 3.2 | 4.7 | 1.4 | Iris-versicolor |
| . . . | | | | |
| 5.7 | 2.8 | 4.1 | 1.3 | Iris-versicolor |
| . . . | | | | |
| 6.3 | 3.3 | 6.0 | 2.5 | Iris-virginica |

# The MySQL table

```
CREATE TABLE IRIS(
Id int primary key auto_increment,
slength decimal(2,1),
swidth decimal(2,1),
plength decimal(2,1),
pwidth decimal(2,1),
class varchar(20)
) ENGINE=INNODB;

LOAD DATA LOCAL INFILE 'DatiIris.csv' INTO TABLE IRIS
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\r\n'
IGNORE 3 LINES
(slength,swidth,plength,pwidth,class);
```

You can use SQL queries to find statistics on the data set.

```
create view IrisSepalSummary as
select  count(*) as N, min(slength) as min_sepal_length,max(slength) as max_sepal_length,
avg(slength) as avg_sepal_length, min(swidth) as min_sepal_width,max(swidth) max_sepal_width,
avg(swidth) as avg_sepal_width from iris;

select * from IrisSepalSummary;

create view IrisPetalSummary as
select  count(*) as N, min(plength) as min_petal_length,max(plength) as max_petal_length,
avg(plength) as avg_petal_length, min(pwidth) as min_petal_width,max(pwidth) max_petal_width,
avg(pwidth) as avg_petal_width from iris;

select * from IrisPetalSummary;

create view SepalSummary as
select class, count(*) as N, min(slength) as min_sepal_length,max(slength) as max_sepal_length,
avg(slength) as avg_sepal_length, min(swidth) as min_sepal_width,max(swidth) max_sepal_width,
avg(swidth) as avg_sepal_width from iris
group by class;

select * from SepalSummary;

create view PetalSummary as
select class, count(*) as N, min(plength) as min_petal_length,max(plength) as max_petal_length,
avg(plength) as avg_petal_length, min(pwidth) as min_petal_width,max(pwidth) max_petal_width,
avg(pwidth) as avg_petal_width from iris
group by class;

select * from PetalSummary;
```

# Weka

- A software for Data Mining written in Java and distributed under the GNU Public License, available at `www.cs.waikato.ac.nz/ml/weka`
  - `Waikato Environment for Knowledge Analysis`
- Used in scientific, didactic and application areas, include:
  - A set of tools for pre-processing, learning algorithms and evaluation methods
  - Graphics Interface
  - A environment to compare the results of learning algorithms

# WEKA Data Management

- The main data type with which WEKA works is the Attribute-Relation file (ARFF file)
- An ARFF file describes the relationship, attributes, and values that it can contain.

```
@RELATION iris
@ATTRIBUTE sepallength REAL
@ATTRIBUTE sepalwidth  REAL
@ATTRIBUTE petallength  REAL
@ATTRIBUTE petalwidth REAL
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}

@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
```

- Another common type of file for `WEKA` is `.csv`.

# MySQL and WEKA

- Connecting WEKA to a MySQL database:
  - you need the driver mysql-connector-java available at http://dev.mysql.com/downloads/connector/j/
  - put it in the archive extension (ext) of Java
  - open WEKA → Explorer → Open DB and specify the following url:
    jdbc:mysql://localhost/DBname
    and the user and password of the database.
- Otherwise, you can directly use a .csv file from WEKA. Use ',' as fields separator and '.' for decimal numbers, an example is file DatiIrisWeka.csv.
- Finally, you can open the file iris.arff under the archive data of Weka

# Preprocessing with Weka

The preprocessing is carried out by means of filters, for example:

- Discretization:
  - Discretize (unsupervised): an instance filter that discretizes a range of numeric attributes in the data set into nominal attributes.
- Normalization:
  - Normalize: normalizes all numeric values in the given data set (apart from the class attribute, if set). The resulting values are by default in $[0, 1]$ for the data used to compute the normalization intervals. But with the scale and translation parameters one can change that, e.g., with $scale = 2.0$ and $translation = -1.0$ you get values in the range $[-1, +1]$
  - Standardize: standardizes all numeric attributes in the given data set to have zero mean and unit variance (apart from the class attribute, if set).

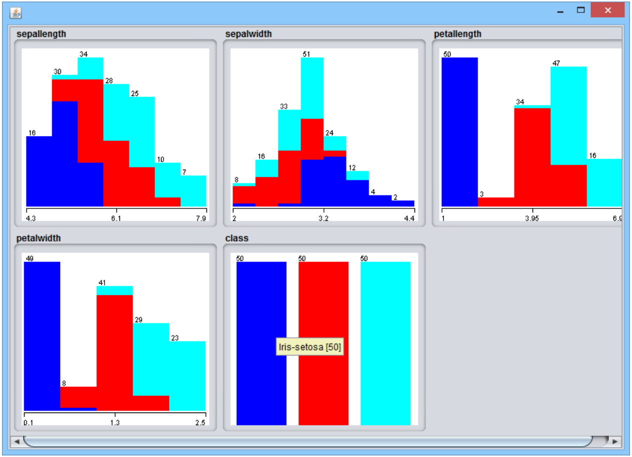
# Preprocessing with Weka

- Sampling:
  - Resample: produces a random subsample of a dataset using either sampling with replacement or without replacement.
- Attribute transformation:
  - NominalToBinary: converts all nominal attributes into binary numeric attributes.
  - AddNoise: an instance filter that changes a percentage of a given attributes values. The attribute must be nominal. Missing value can be treated as value itself.
- Missing values:
  - ReplaceMissingValues: replaces all missing values for nominal and numeric attributes in a data set with the modes and means from the training data.

The preprocessing tab also allow you to visualize data distributions with respect to the classification attribute or other attribute.

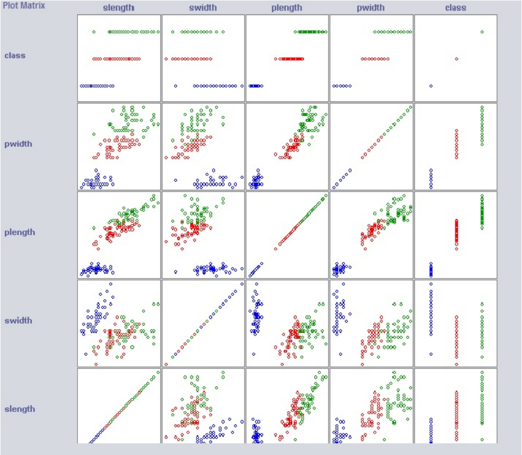
# Visualize Iris data with Weka

Obtained with the `Preprocess` environment and by using `Visualize All`.



# Scatter plots (obtained with Weka)

Scatter plots visualize two variables in a two-dimensional plot. Each axes corresponds to one variable. The colors are Iris-setosa, Iris-versicolor, Iris-virginica



# A note on scatter plots

Data objects with the same values cannot be distinguished in a scatter plot. To avoid this effect, jitter is used, i.e. before plotting the points, small random values are added to the coordinates. Jitter is essential for categorical attributes.

# Reminder: arithmetic mean, variance, standard deviation, median

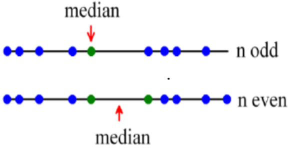
- Arithmetic mean:

$$mean(x) = \bar{x} = \frac{1}{n}\sum_{k=1}^{n} x_k$$

  (sensitive to the presence of outliers)

- Variance: $var(x) = \frac{1}{n}\sum_{k=1}^{n}(x_k - \bar{x})^2$
- Standard deviation: $\sigma = \sqrt{var(x)}$
- Median: the value in the middle (for the values given in increasing order):

$$median(x) = \begin{cases} x_{m+1} \; if \;\; n = 2m+1 \\ (x_m + x_{m+1})/2 \; if \;\; n = 2m \end{cases}$$



# Reminder: frequency, mode

- The frequency of an attribute value is the percentage of time the value occurs in the data set. For example, given the attribute gender and a representative population of people, the gender female occurs about 50% of the time.
- The mode of a an attribute is the most frequent attribute value. The notions of frequency and mode are typically used with categorical data.
- For the iris data sete, the three types of flowers all have the same frequency and therefore the notion of a mode is not interesting.

# Reminder: quantiles, quartiles, interquartile range

- $q\%$-quantile ($0 < q < 100$): the value for which $q\%$ of the values are smaller and $100 - q\%$ are larger.
- The median is the 50%-quantile.
- Quartiles: 25%-quantile (1st quartile), median (2nd quantile), 75%-quantile (3rd quartile).
- Interquartile range (IQR): 3rd quantile - 1st quantile.

# Data understanding with R

R Code accompanying the book *Introduction to Data Mining* by Tan, Steinbach and Kumar can be found at `https://github.com/mhahsler/Introduction_to_Data_Mining_R_Examples`

```
> iris <- datasets::iris
> summary(iris)
```

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
       Species  
 setosa    :50  
 versicolor:50  
 virginica :50  
```

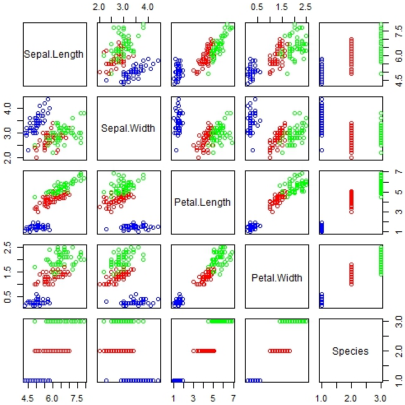
The `summary()` function gives summary statistics for any dataset. It can also be called on one variable instead of on the whole dataset.

```
> summary(iris$Sepal.Length)
```

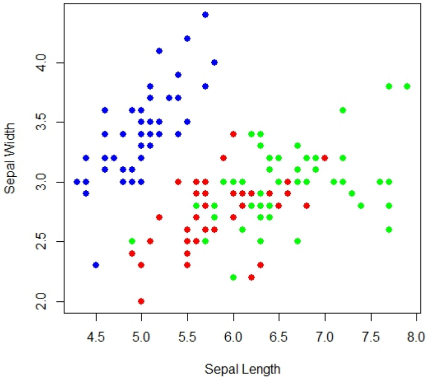
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  4.300   5.100   5.800   5.843   6.400   7.900 
```

# Scatter plots with R

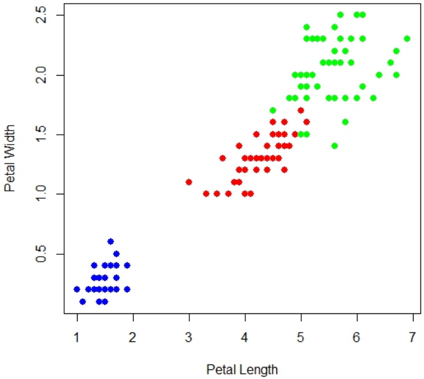
```
> iris <- datasets::iris
> iris2 <- iris[,-5]
> species_labels <- iris[,5]
> colors <- c("blue","red", "green")
> species_col <- colors[as.numeric(species_labels)]
> plot(iris,col = species_col)
```



```
> SepalWidth<-iris[,2]
> SepalLength<-iris[,1]
> PetalWidth<-iris[,4]
> PetalLength<-iris[,3]
> plot(SepalLength,SepalWidth,col = species_col, pch=19,cex = 1.1,
+   xlab="Sepal Length",ylab="Sepal Width")
```

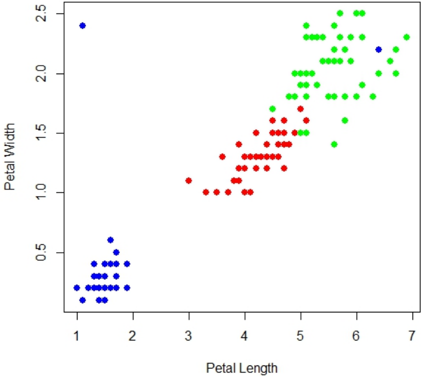


```
> plot(PetalLength,PetalWidth,col = species_col,pch=19,cex = 1.1,
+   xlab="Petal Length",ylab="Petal Width")
```



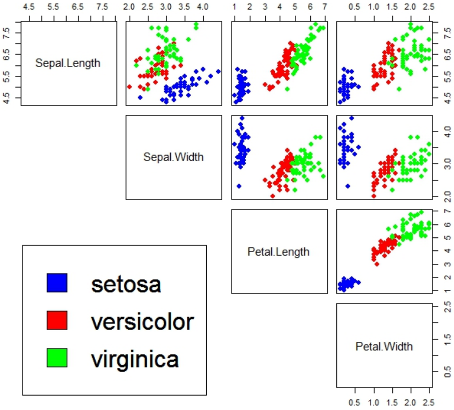
The two attributes petal length and width provide a better separation of the classes Iris versicolor and Iris virginica than the sepal length and width.

```
> plot(PetalLength0,PetalWidth0,col = species_col0,pch=19,cex = 1.1,
+   xlab="Petal Length",ylab="Petal Width")
```



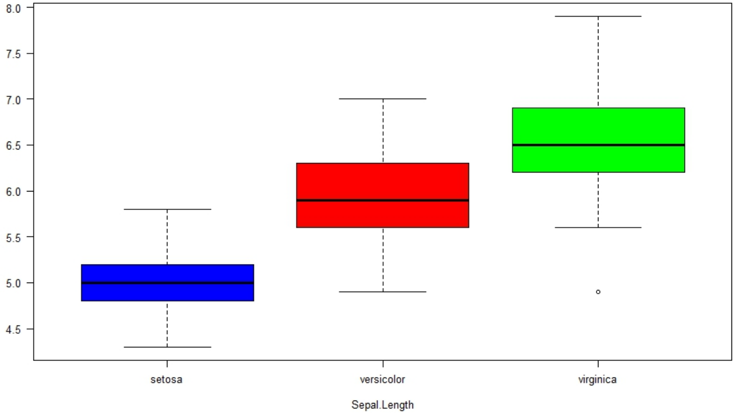
The Iris data set with two (additional artificial) outliers. One is an outlier for the whole data set, one for the class Iris setosa.

```
> pairs(iris2, col = species_col,
+       lower.panel = NULL,
+       cex.labelsiris=2, pch=19, cex = 1.2)
> par(xpd = TRUE)
> legend(x = 0.05, y = 0.4, cex = 2,
+ legend = as.character(levels(species_labels)),
+ fill = unique(species_col))
> par(xpd = NA)
```



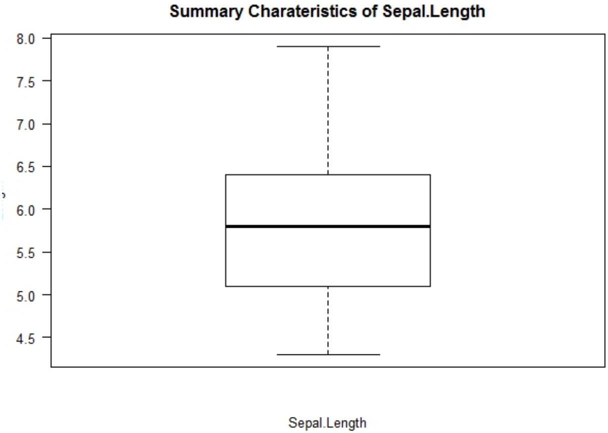
# Boxplots with R

```
> boxplot(Sepal.Length~Species,data = iris,xlab="Sepal.Length",col=c("blue","red", "green"))
```



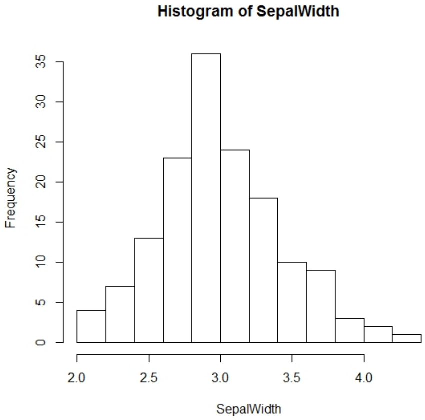
The median and the interquartile range are shown.

```
> boxplot(iris[,1],xlab="Sepal.Length",ylab="Length",main="Summary Charateristics of Sepal.Length")
```



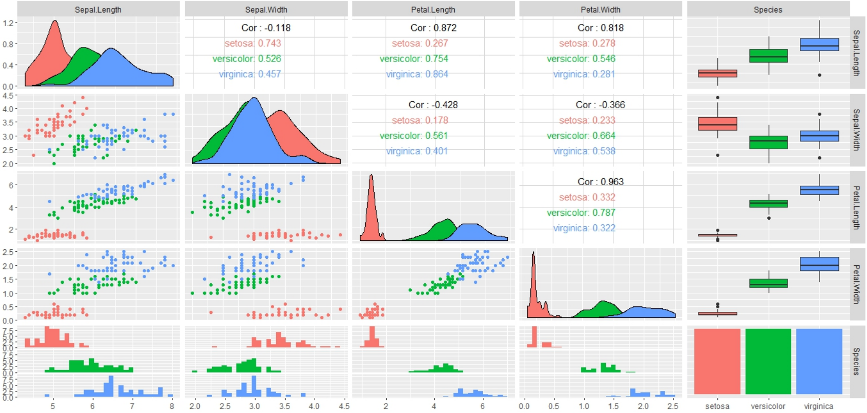
# Histograms with R

```
> hist(SepalWidth)
```



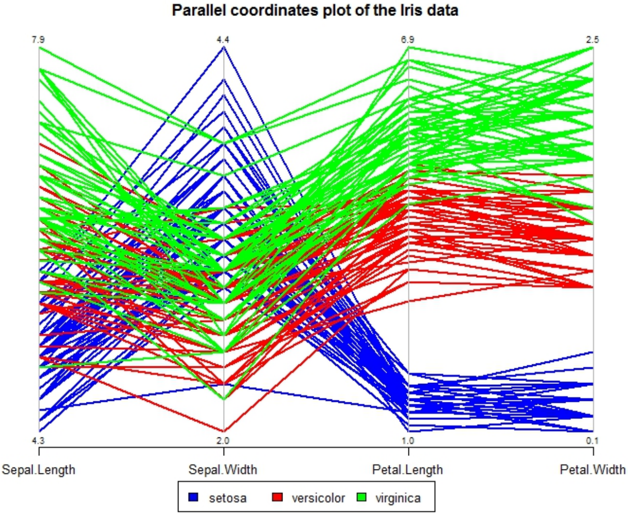
# Alternative scatter plot matrix

```
> library("GGally")
> ggpairs(iris,  ggplot2::aes(colour=Species))
```



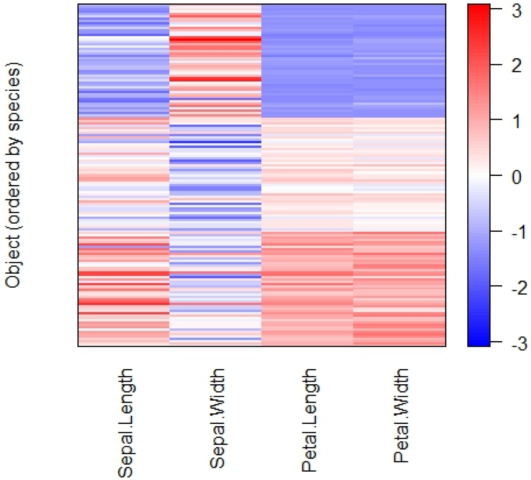
# Parallel coordinates plot of the data

```
> par(las = 1, mar = c(4.5, 3, 3, 2) + 0.1, cex = .8)
> MASS::parcoord(iris2, col = species_col1, var.label = TRUE, lwd = 2)
# Add Title
> title("Parallel coordinates plot of the Iris data")
# Add a legend
> par(xpd = TRUE)
> legend(x = 1.75, y = -.13, cex = 1,
+ legend = as.character(levels(species_labels)),
+ fill = unique(species_col1), horiz = TRUE)
> par(xpd = NA)
```


Parallel coordinates plot of the Iris data
7.9
4.4
6.9
2.5
4.3
2.0
1.0
0.1
Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
setosa
versicolor
virginica

# Visualization of the Iris Data Matrix

```
> iris_matrix <- as.matrix(iris[,1:4])
> library(seriation) ## for pimage
> iris_scaled <- scale(iris_matrix)
# values smaller than the average are blue
# and larger ones are red
> pimage(iris_scaled,
+   ylab="Object (ordered by species)",
+ main="Standard deviations from the feature mean")
```

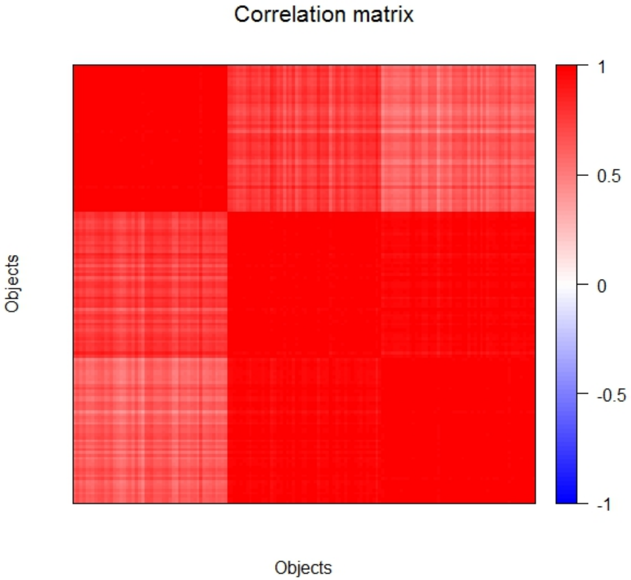


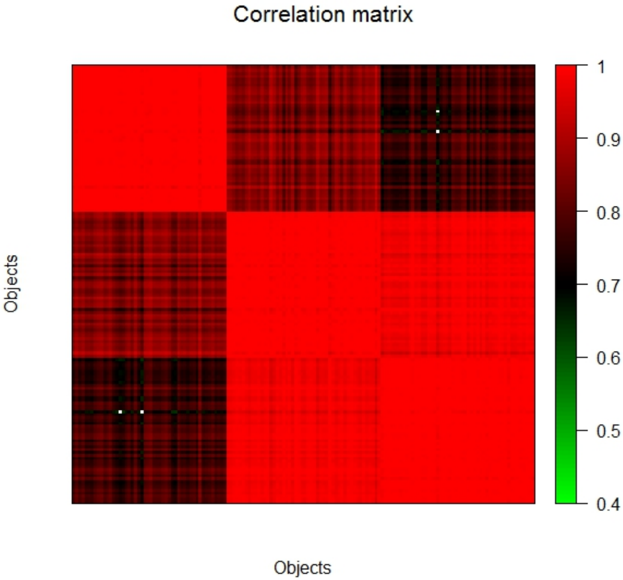
Standard deviations from the feature mean

# Visualize correlation between objects (via correlation matrix)

```
> iris_matrix <- as.matrix(iris[,1:4])
> library(seriation) ## for pimage
# Correlation between objects
> cm2 <- cor(t(iris_matrix))
> pimage(cm2,
+ main="Correlation matrix", xlab="Objects", ylab="Objects",
+   zlim = c(-1,1),col = bluered(100))

> pimage(cm2,
+ main="Correlation matrix", xlab="Objects", ylab="Objects",
+   zlim = c(0.4,1),col = greenred(100))
```

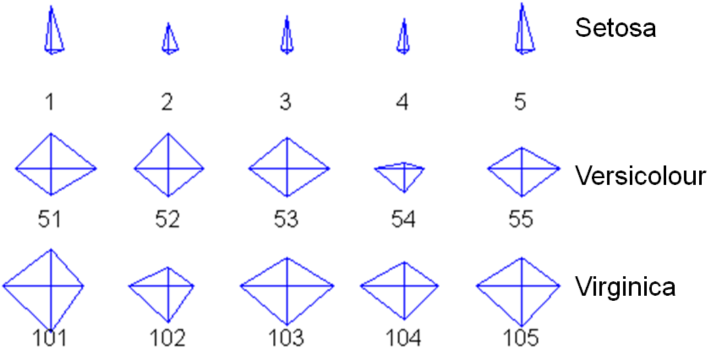

Correlation matrix
Objects
Objects
1
0.5
0
-0.5
-1


Correlation matrix
Objects
Objects
1
0.9
0.8
0.7
0.6
0.5
0.4

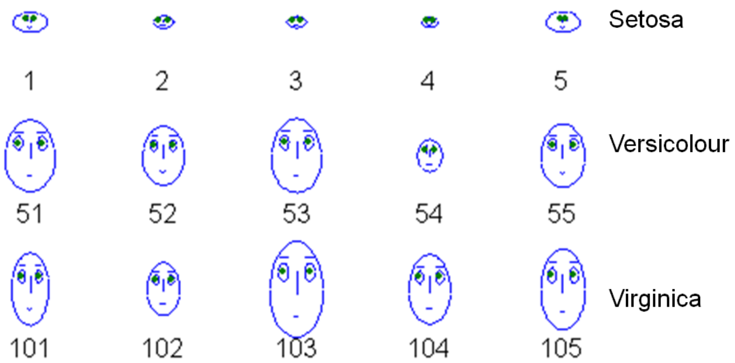
# Other Visualization Techniques

- Star Plots: this technique uses one axis for each attribute, the axes radiate from a central point. The line connecting the values of an object is a polygon
- Chernoff Faces: approach created by Herman Chernoff, associates each attribute with a characteristic of a face; the values of each attribute determine the appearance of the corresponding facial characteristic:
  - sepal lenght=size of face
  - sepal width= forehead/jaw relative arc length
  - petal length= shape of forehead
  - petal width=shape of jaw

# Star Plots for Iris Data



# Chernoff Faces for Iris Data



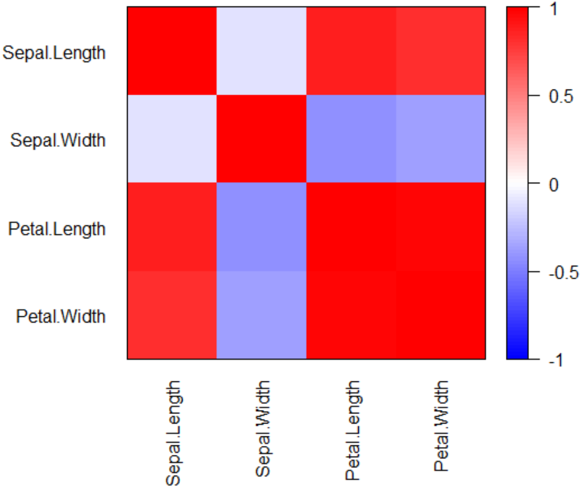
# Proving similiar behaviour of attributes

```
> Pearsoncorrelation<-cor(iris2,method="pearson")
> Pearsoncorrelation

             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000

> pimage(Pearsoncorrelation)
```

# . . . and the corresponding visualization



```
> Spearmancorrelation<-cor(iris2,method="spearman")
> Spearmancorrelation

             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000  -0.1667777    0.8818981   0.8342888
Sepal.Width    -0.1667777   1.0000000   -0.3096351  -0.2890317
Petal.Length    0.8818981  -0.3096351    1.0000000   0.9376668
Petal.Width     0.8342888  -0.2890317    0.9376668   1.0000000
```