R Code for Chapter 2 of Introduction to Data Mining: Data (Exploring Data)

Michael Hahsler

2021-07-11

- Basic statistics
- Tabulate data
- Percentiles
- Visualizations
    - Histogram
    - Scatter plot
    - Scatter plot matrix
    - Boxplot
    - ECDF: Empirical Cumulative Distribution Function
    - Data matrix visualization
    - Correlation matrix
    - Parallel coordinates plot

This is additional code related to chapter 2 of *“Introduction to Data Mining”* by Pang-Ning Tan, Michael Steinbach and Vipin Kumar. **See table of contents (https://github.com/mhahsler/Introduction_to_Data_Mining_R_Examples#readme) for code examples for other chapters.**

Load the iris data set ##### Dingyi Duan #### ##### University of San Diego #### ##### ADS 502 #### ##### Module 2 ####

```
data(iris)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Basic statistics

Get summary statistics

```
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  
##                 
##                 
##                 
```

Get mean and standard deviation for sepal length

```
mean(iris$Sepal.Length)
```

```
## [1] 5.843333
```

```
sd(iris$Sepal.Length)
```

```
## [1] 0.8280661
```

Ignor missing values (Note: this data does not contain any, but this is what you would do)

```
mean(iris$Sepal.Length, na.rm = TRUE)
```

```
## [1] 5.843333
```

Robust mean (trim 10% of observations from each end of the distribution)

```
mean(iris$Sepal.Length, trim = .1)
```

```
## [1] 5.808333
```

Apply mean, sd and median to columns (MARGIN=2)

```
apply(iris[1:4], MARGIN=2, mean)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width 
##     5.843333     3.057333     3.758000     1.199333 
```

```
apply(iris[1:4], MARGIN=2, median)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width 
##         5.80         3.00         4.35         1.30
```

```
apply(iris[1:4], MARGIN=2, sd)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width 
##    0.8280661    0.4358663    1.7652982    0.7622377
```

```
apply(iris[1:4], MARGIN=2, var)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width 
##    0.6856935    0.1899794    3.1162779    0.5810063
```

```
apply(iris[1:4], MARGIN=2, min)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width 
##          4.3          2.0          1.0          0.1
```

```
apply(iris[1:4], MARGIN=2, max)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width 
##          7.9          4.4          6.9          2.5
```

Define your own statistic: E.g., MAD (median absolute deviation)

```
mad <- function(x) median(abs(x-mean(x)))
apply(iris[1:4], MARGIN=2, mad)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width 
##    0.6566667    0.2573333    1.7920000    0.7993333
```

Tabulate data

Discretize the data first since there are too many values (cut divides the range by breaks, see package discretization for other methods)

```
iris_discrete <- data.frame(
    Sepal.Length= cut(iris$Sepal.Length, breaks=3,
        labels=c("small", "medium", "large"), ordered=TRUE),
    Sepal.Width= cut(iris$Sepal.Width, breaks=3,
        labels=c("small", "medium", "large"), ordered=TRUE),
    Petal.Length= cut(iris$Petal.Length, breaks=3,
        labels=c("small", "medium", "large"), ordered=TRUE),
    Petal.Width= cut(iris$Petal.Width, breaks=3,
        labels=c("small", "medium", "large"), ordered=TRUE),
    Species = iris$Species
    )

head(iris_discrete)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1        small      medium        small       small  setosa
## 2        small      medium        small       small  setosa
## 3        small      medium        small       small  setosa
## 4        small      medium        small       small  setosa
## 5        small      medium        small       small  setosa
## 6        small       large        small       small  setosa
```

```
summary(iris_discrete)
```

```
##  Sepal.Length Sepal.Width Petal.Length Petal.Width       Species
##  small :59    small :47   small :50    small :50   setosa    :50
##  medium:71    medium:88   medium:54    medium:54   versicolor:50
##  large :20    large :15   large :46    large :46   virginica :50
```

Create some tables

```
table(iris_discrete$Sepal.Length, iris_discrete$Sepal.Width)
```

```
##
##           small medium large
##   small     12     37    10
##   medium    31     37     3
##   large      4     14     2
```

```
table(iris_discrete$Petal.Length, iris_discrete$Petal.Width)
```

```
##
##           small medium large
##   small     50      0     0
##   medium     0     48     6
##   large      0      6    40
```

```
table(iris_discrete$Petal.Length, iris_discrete$Species)
```

```
##
##          setosa versicolor virginica
##   small      50          0         0
##   medium      0         48         6
##   large       0          2        44
```

```
#table(iris_discrete)
```

Test if the two features are independent given the counts in the contingency table (H0: independence)

p-value: the probability of seeing a more extreme value of the test statistic under the assumption that H0 is correct. Low p-values (typically less than .05 or .01) indicate that H0 should be rejected.

```
tbl <- table(iris_discrete$Sepal.Length, iris_discrete$Sepal.Width)
tbl
```

```
##
##          small medium large
##   small     12     37    10
##   medium    31     37     3
##   large      4     14     2
```

```
chisq.test(tbl)
```

```
## Warning in chisq.test(tbl): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 12.879, df = 4, p-value = 0.01188
```

Fisher’s exact test is better for small counts (cells with counts <5)

```
fisher.test(tbl)
```

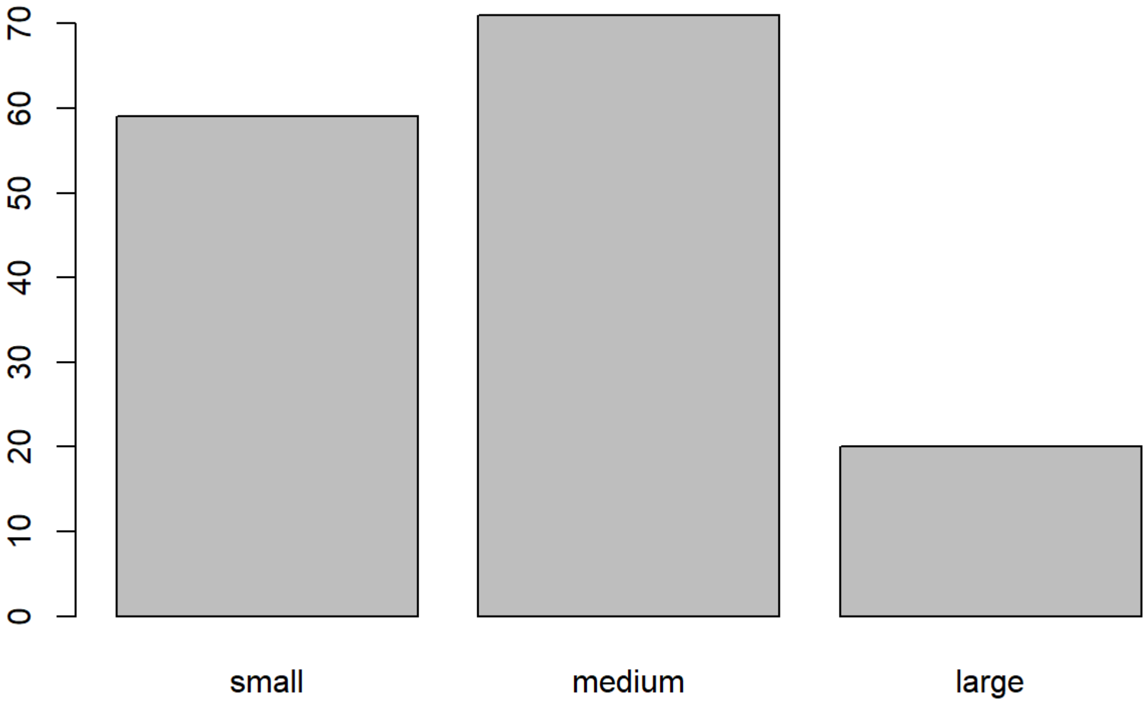
```
##
##  Fisher's Exact Test for Count Data
##
## data:  tbl
## p-value = 0.01063
## alternative hypothesis: two.sided
```

Plot the distribution for a discrete variable

```
table(iris_discrete$Sepal.Length)
```

```
##
##  small medium  large
##     59     71     20
```

```
barplot(table(iris_discrete$Sepal.Length))
```



Percentiles

```
apply(iris[1:4], MARGIN=2, quantile)
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width
## 0%            4.3         2.0         1.00         0.1
## 25%           5.1         2.8         1.60         0.3
## 50%           5.8         3.0         4.35         1.3
## 75%           6.4         3.3         5.10         1.8
## 100%          7.9         4.4         6.90         2.5
```

Interquartile range

```
quantile(iris$Petal.Length)
```

```
##   0%  25%  50%  75% 100% 
## 1.00 1.60 4.35 5.10 6.90
```

```
quantile(iris$Petal.Length)[4] - quantile(iris$Petal.Length)[2]
```

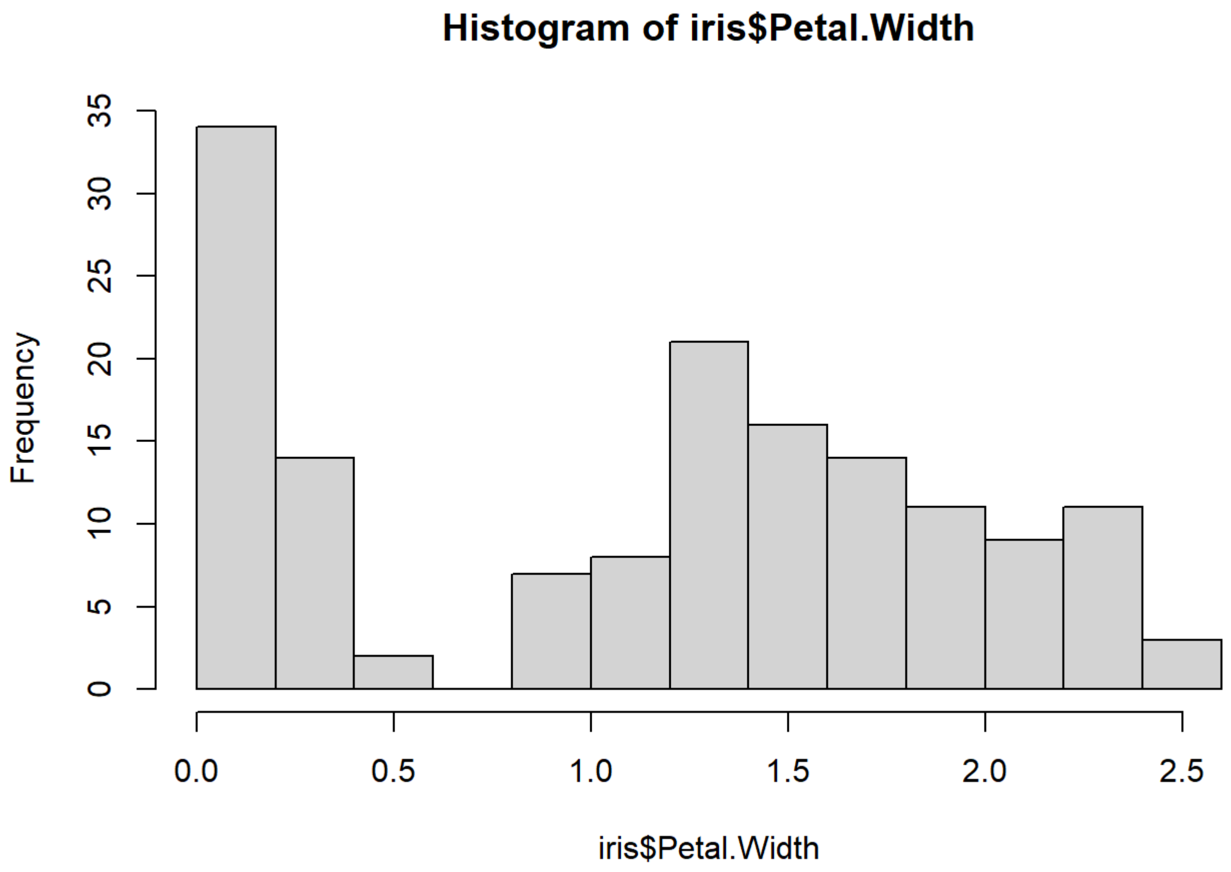
```
## 75% 
## 3.5
```

Visualizations

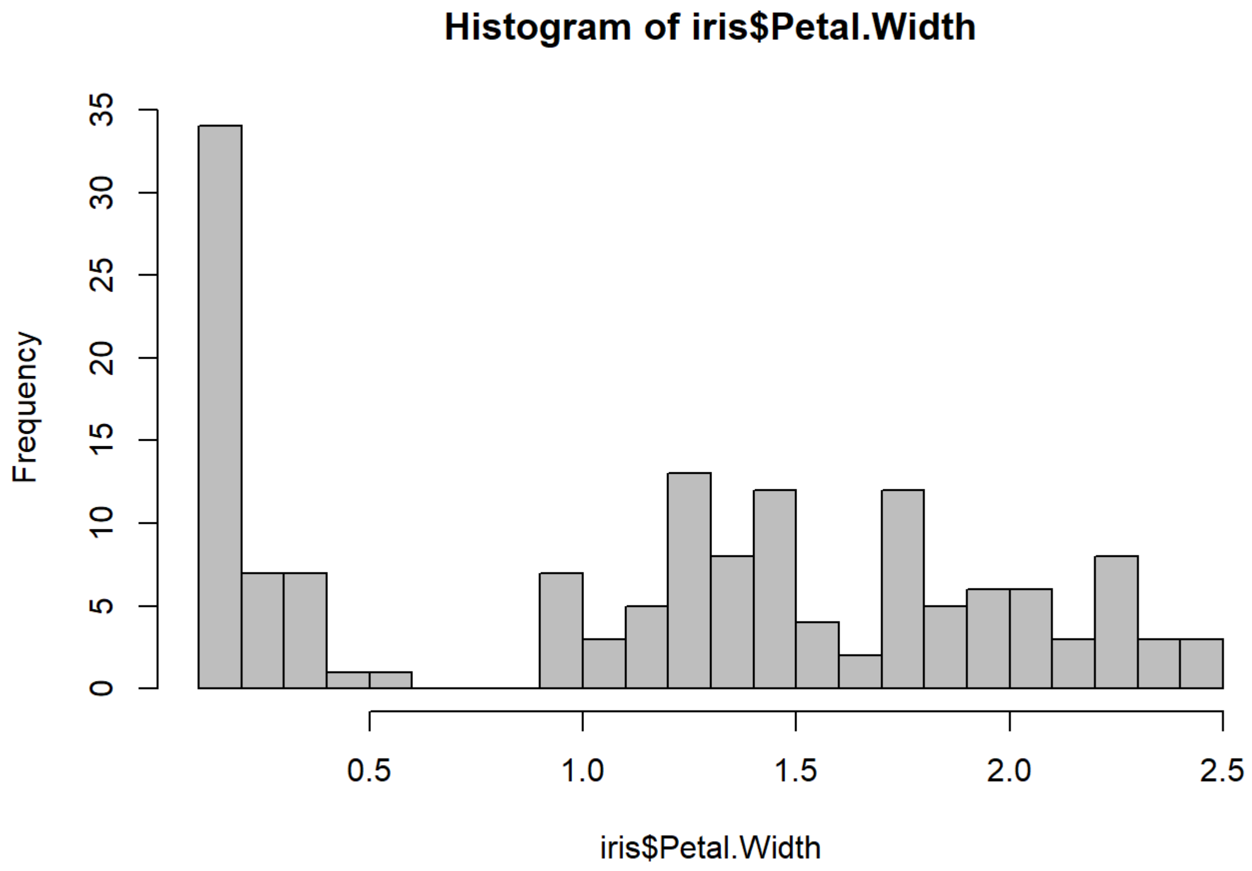
Histogram

Show the distribution of a single numeric variable

```
hist(iris$Petal.Width)
```



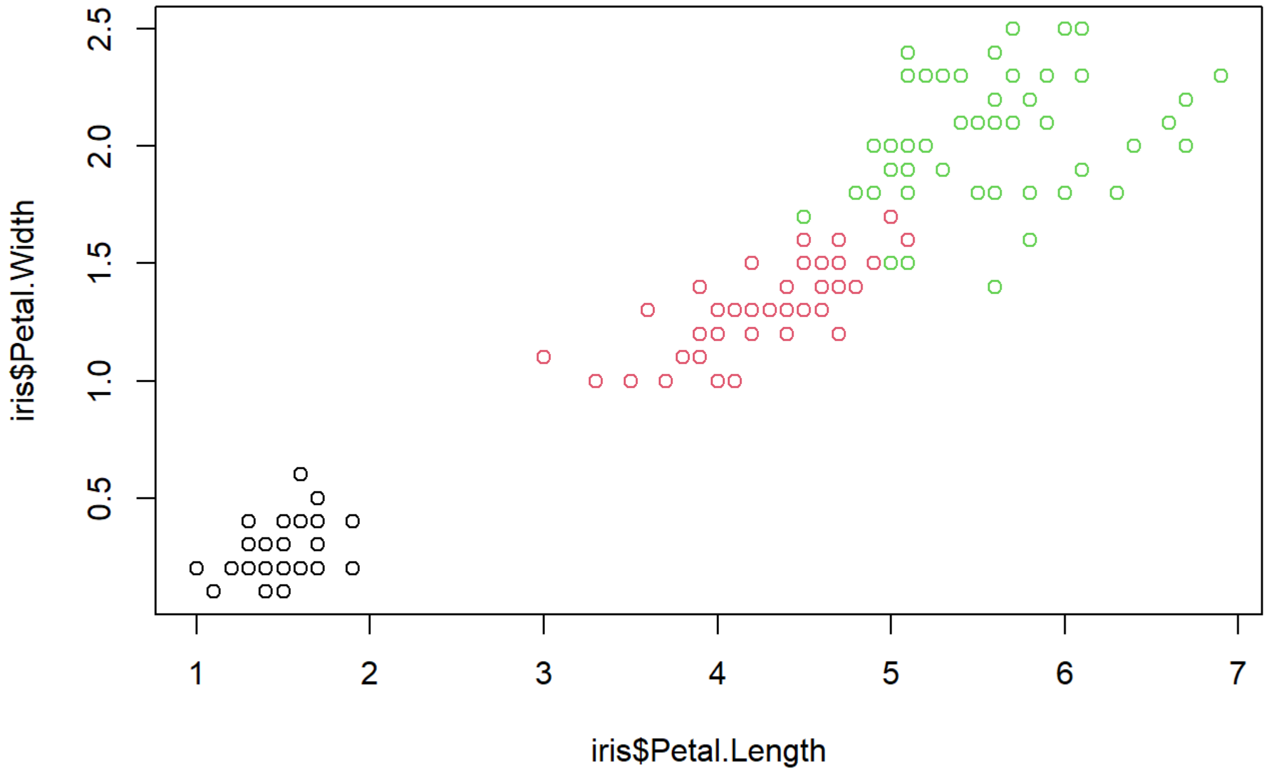
```
hist(iris$Petal.Width, breaks=20, col="grey")
```



Scatter plot

Show the relationship between two numeric variables

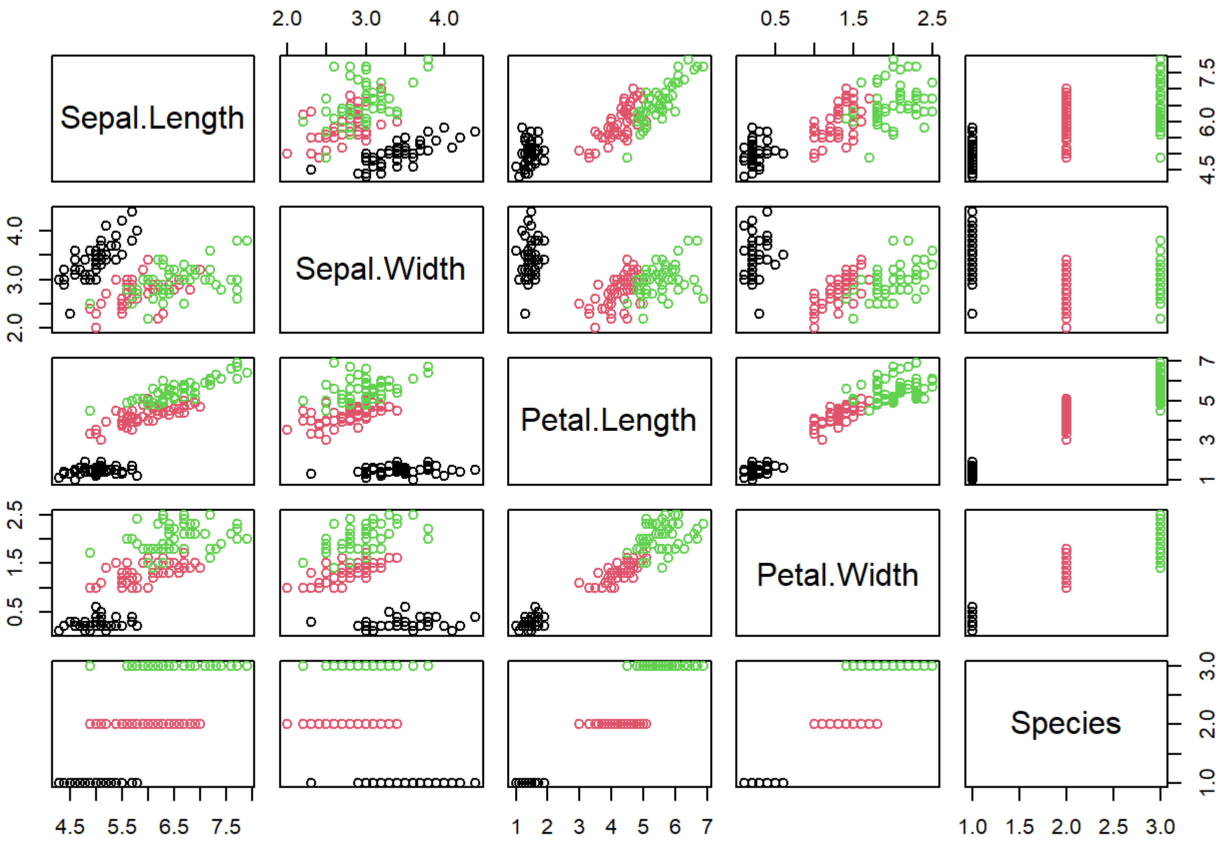
```
plot(x=iris$Petal.Length, y=iris$Petal.Width, col=iris$Species)
```



Scatter plot matrix

Show the relationship between several numeric variables

```
pairs(iris, col=iris$Species)
```



Alternative scatter plot matrix

```
library("GGally")
```

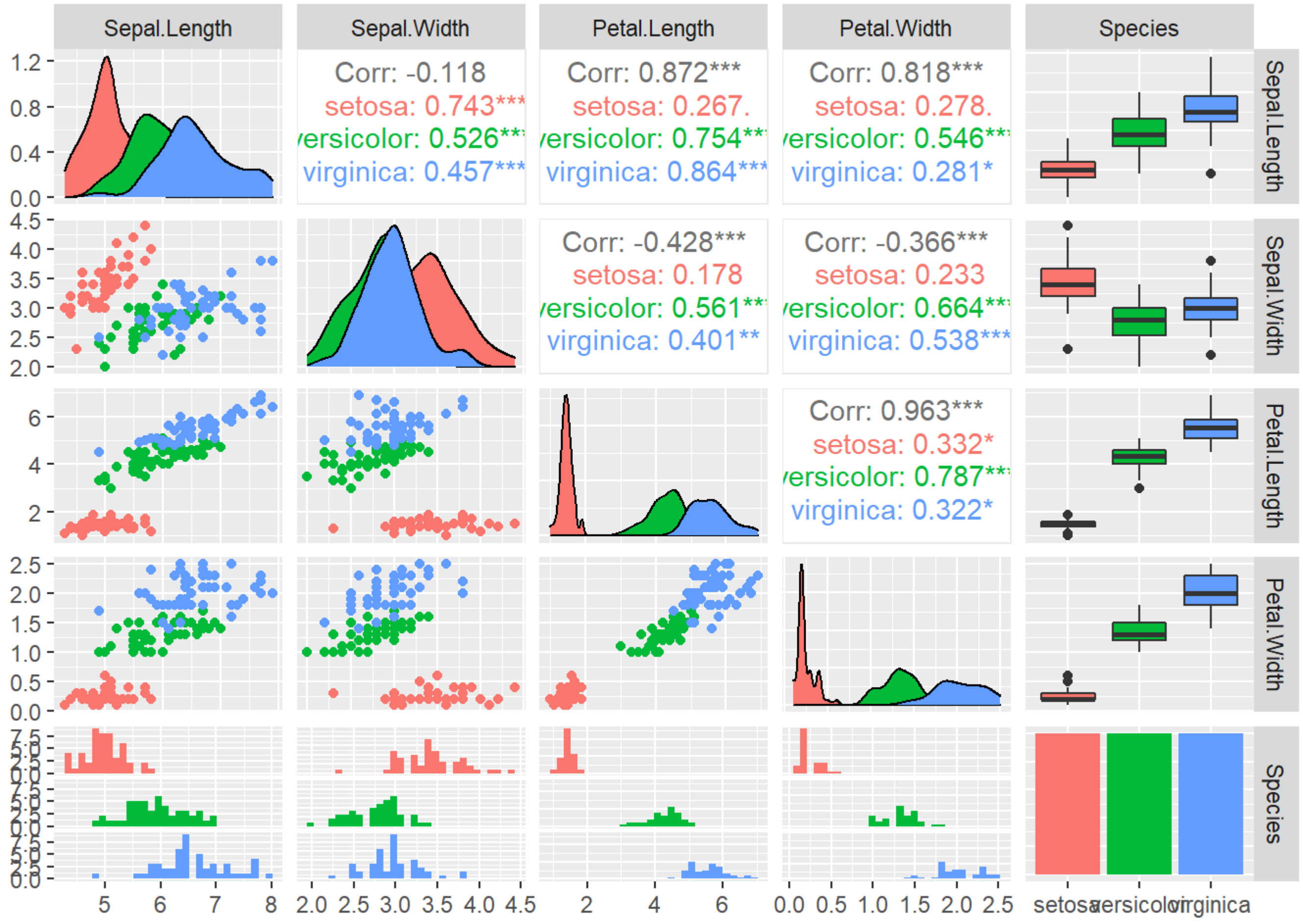
```
## Loading required package: ggplot2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
ggpairs(iris,  ggplot2::aes(colour=Species))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

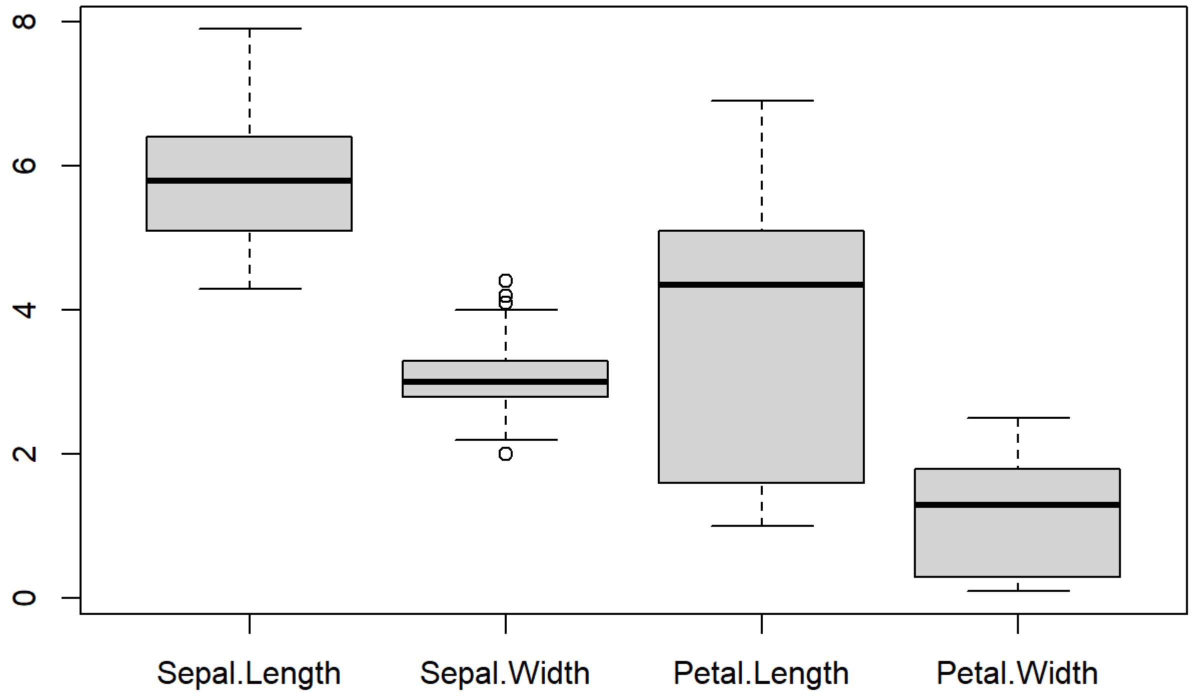
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Boxplot

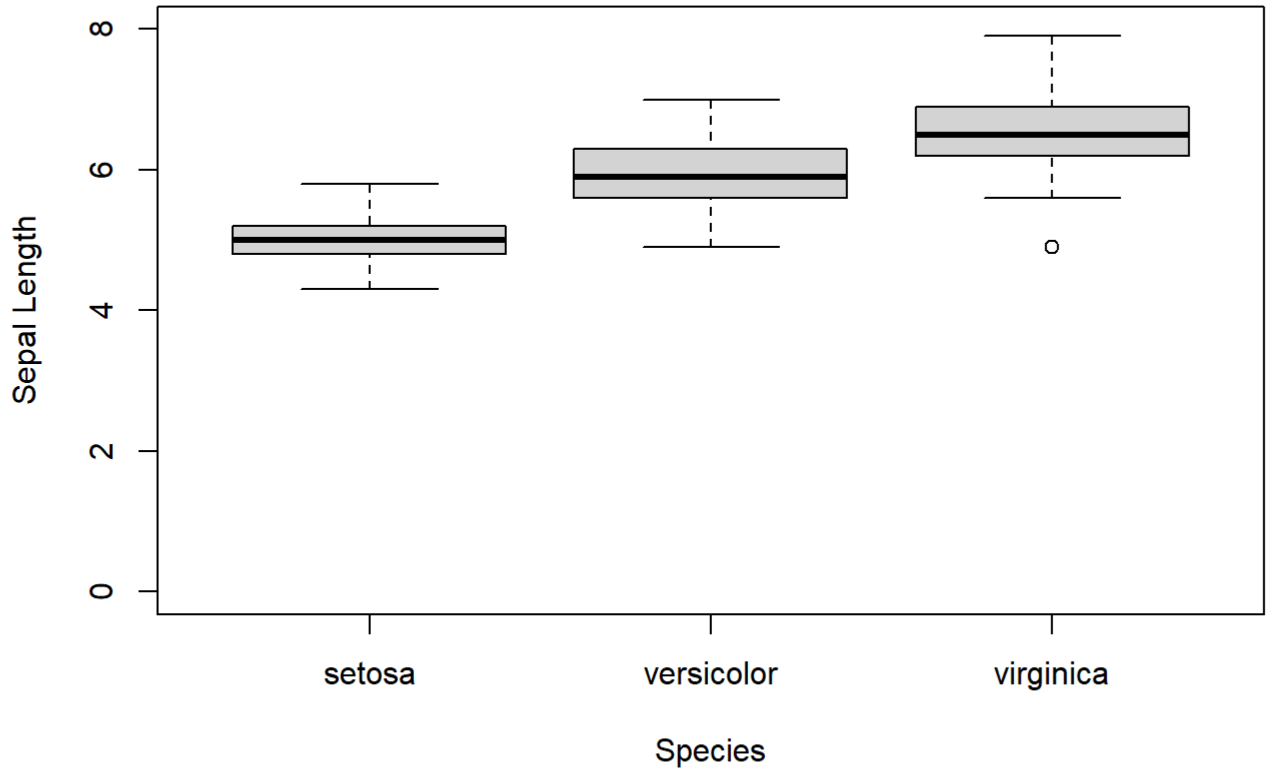
Compare the distribution of several continuous variables

```
boxplot(iris[,1:4])
```



Compare the distribution of a single continuous variables grouped by a nominal variable

```
boxplot(Sepal.Length ~ Species, data = iris,
  ylab = "Sepal Length", ylim = c(0,8))
```



Group-wise averages

```
aggregate(Sepal.Length ~ Species, data=iris, FUN = mean)
```

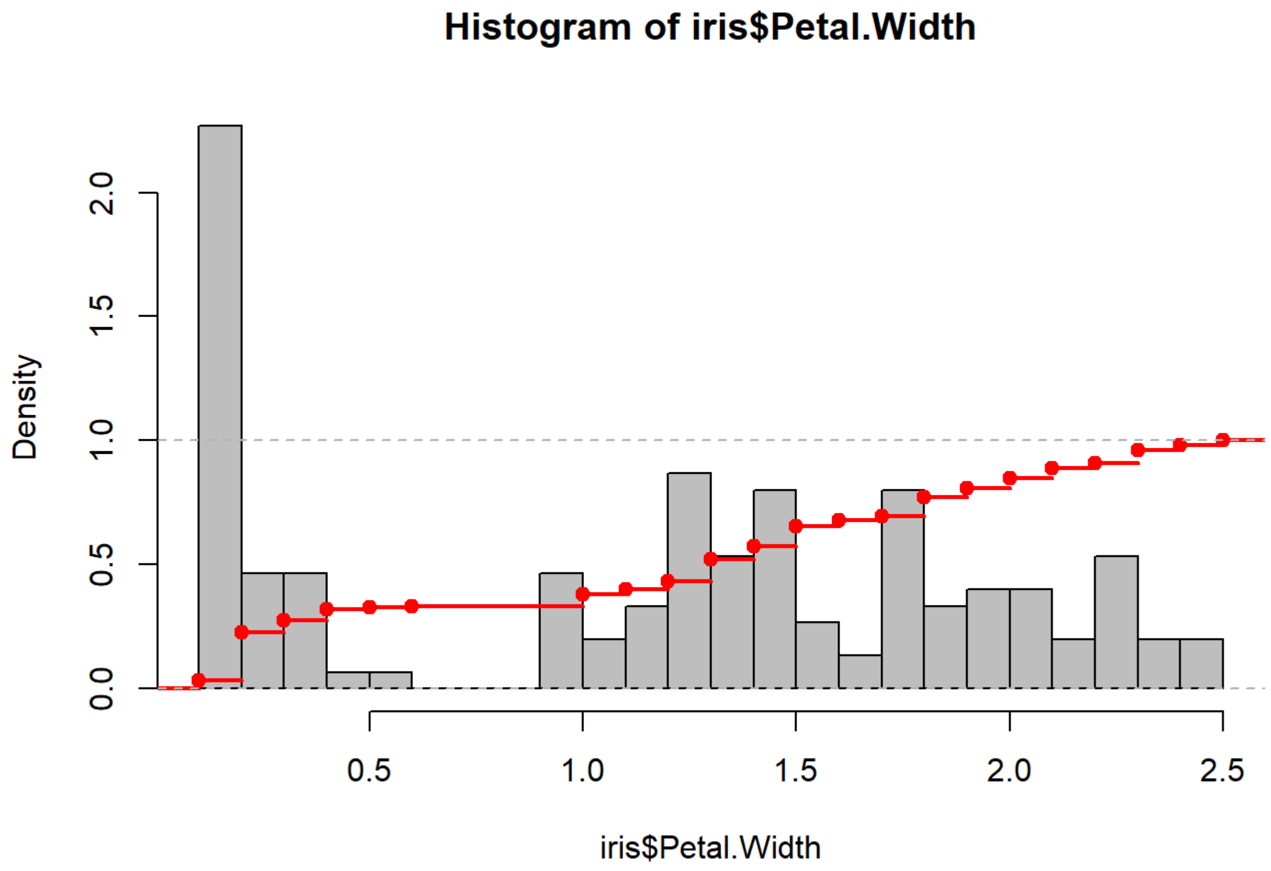
```
##      Species Sepal.Length
## 1     setosa        5.006
## 2 versicolor        5.936
## 3  virginica        6.588
```

```
aggregate(Sepal.Width ~ Species, data=iris, FUN = mean)
```

```
##      Species Sepal.Width
## 1     setosa       3.428
## 2 versicolor       2.770
## 3  virginica       2.974
```

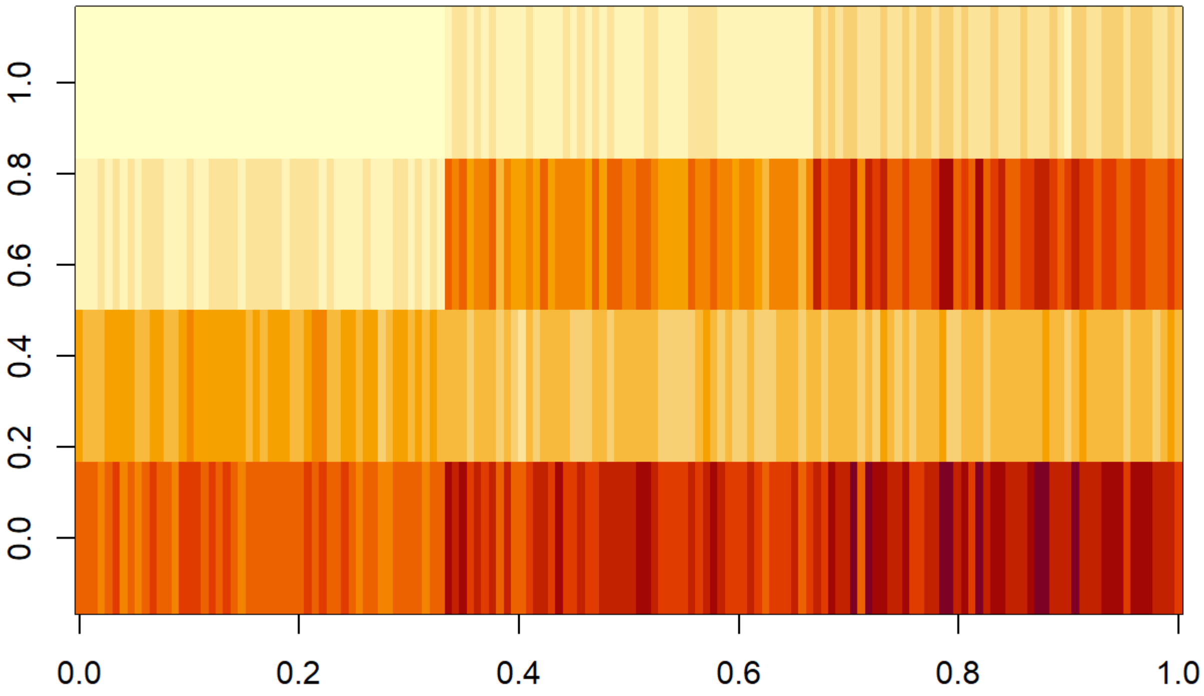
ECDF: Empirical Cumulative Distribution Function

```
e <- ecdf(iris$Petal.Width)
hist(iris$Petal.Width, breaks=20, freq=FALSE, col="gray")
lines(e, col="red", lwd=2)
```

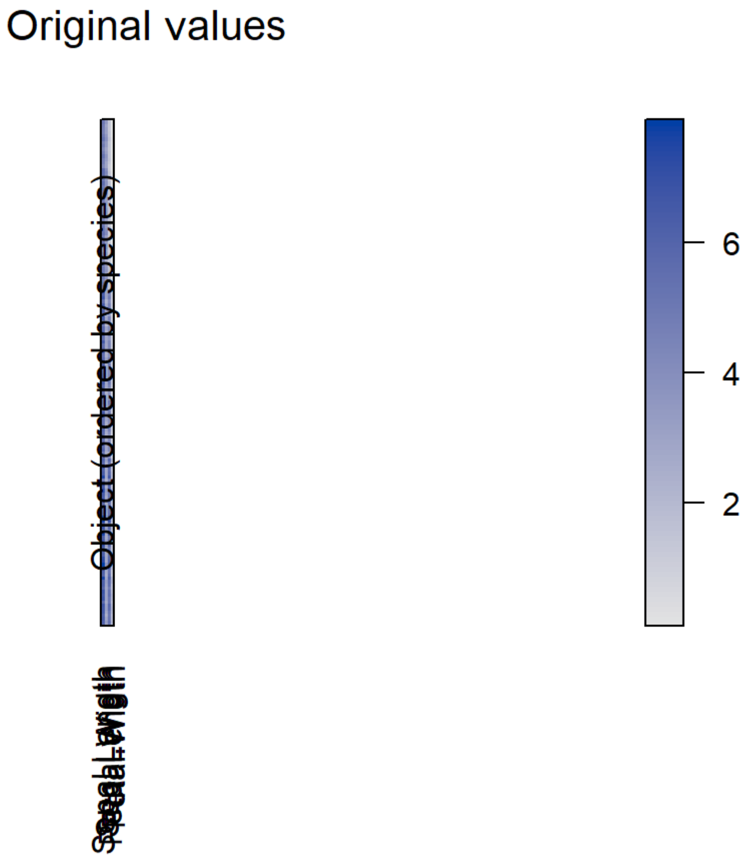


Data matrix visualization

```
iris_matrix <- as.matrix(iris[,1:4])
image(iris_matrix)
```

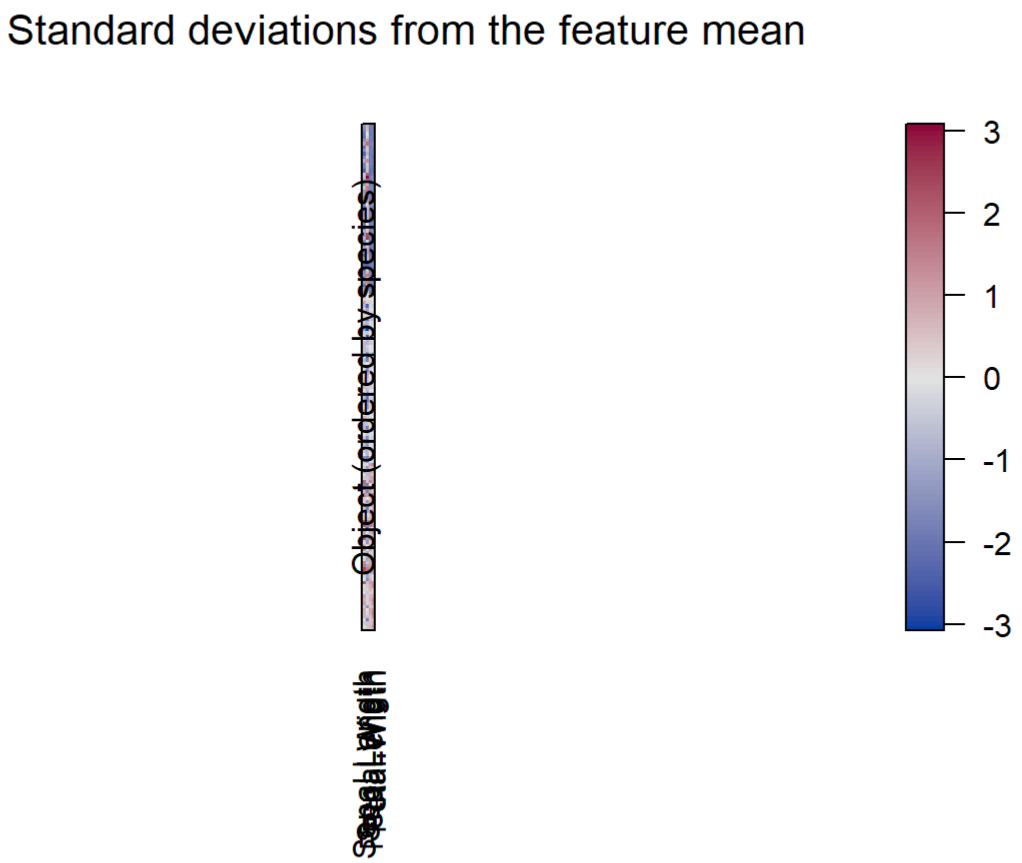


```
library(seriation) ## for pimage
pimage(iris_matrix, ylab="Object (ordered by species)",
  main="Original values", colorkey=TRUE)
```



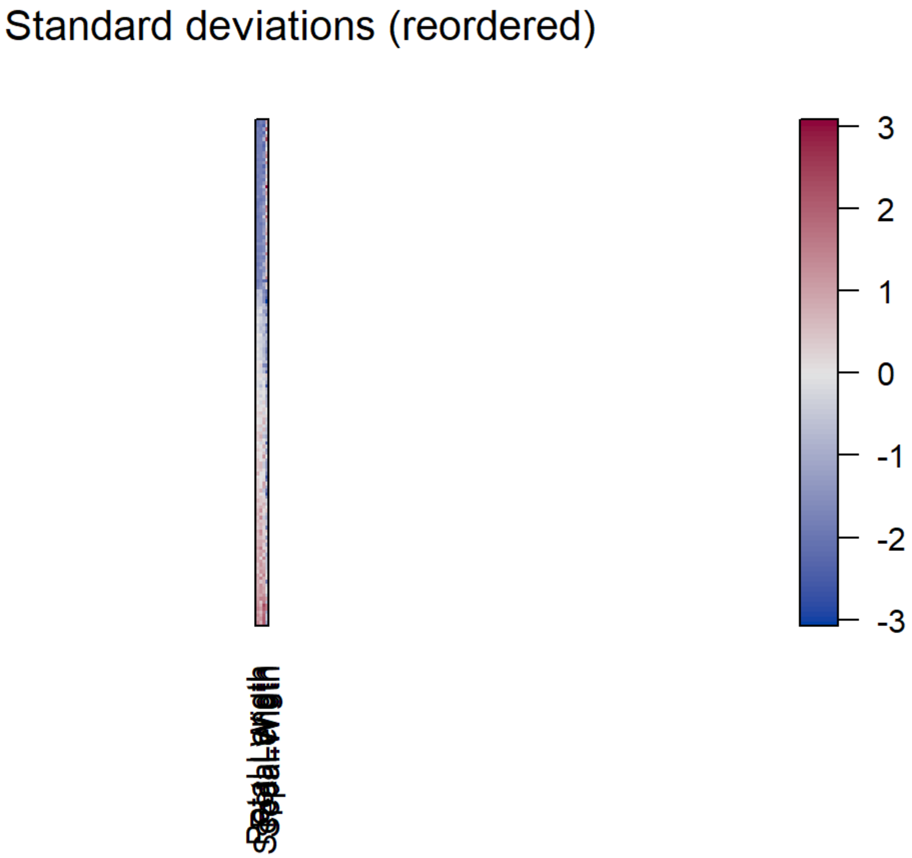
values smaller than the average are blue and larger ones are red

```
iris_scaled <- scale(iris_matrix)
pimage(iris_scaled,
  ylab="Object (ordered by species)",
    main="Standard deviations from the feature mean")
```



use reordering of features and objects

```
pimage(iris_scaled, order = seriate(iris_scaled),
  main="Standard deviations (reordered)")
```



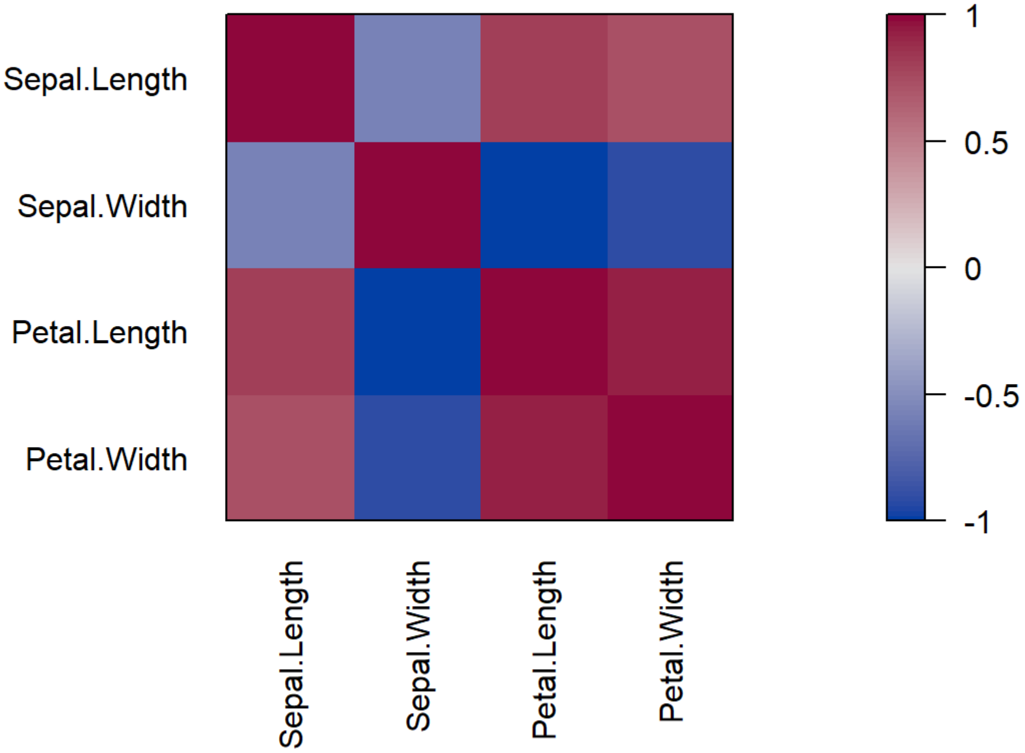
Correlation matrix

Calculate and visualize the correlation between features

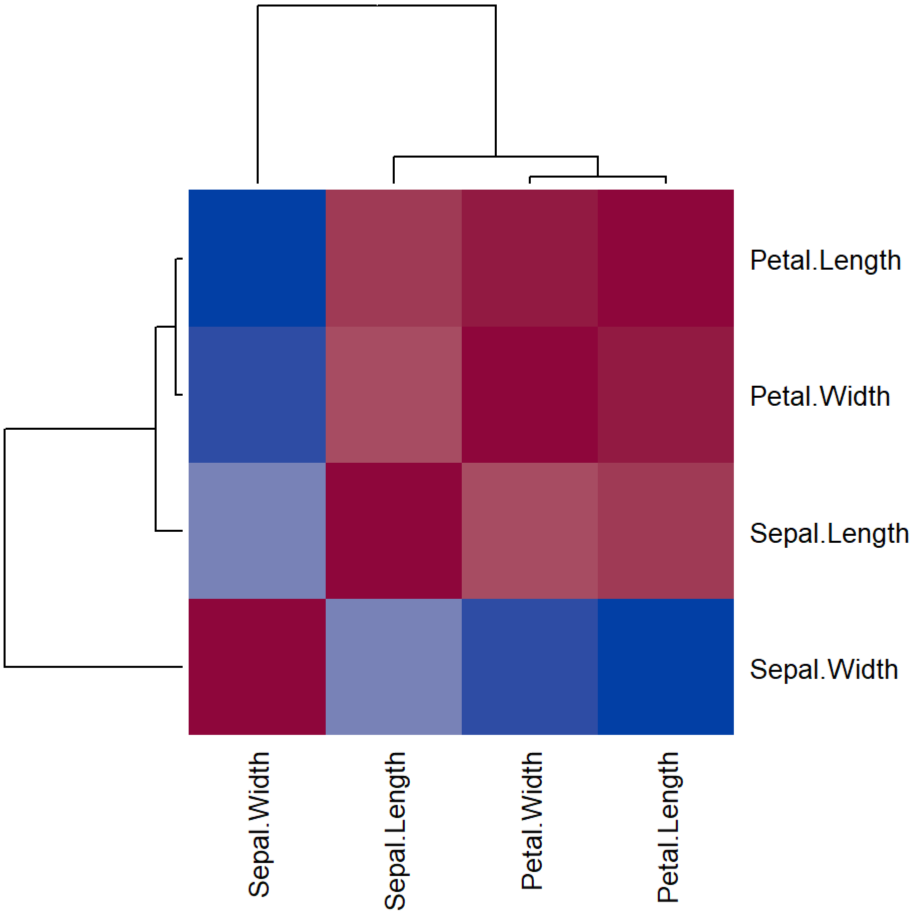
```
cm1 <- cor(iris_matrix)
cm1
```

```
##              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
```

```
library(seriation) ## for pimage and hmap
pimage(cm1)
```



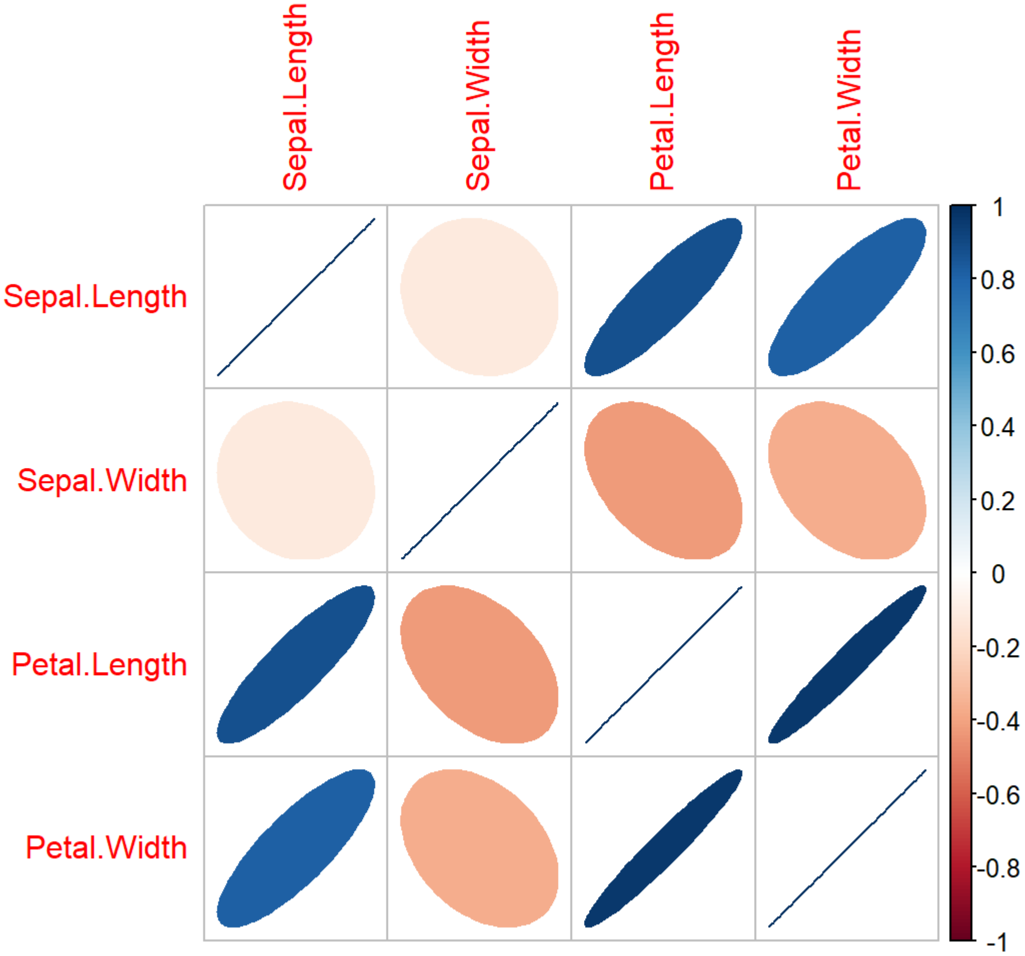
```
hmap(cm1, margin = c(7,7), cexRow = 1, cexCol = 1)
```



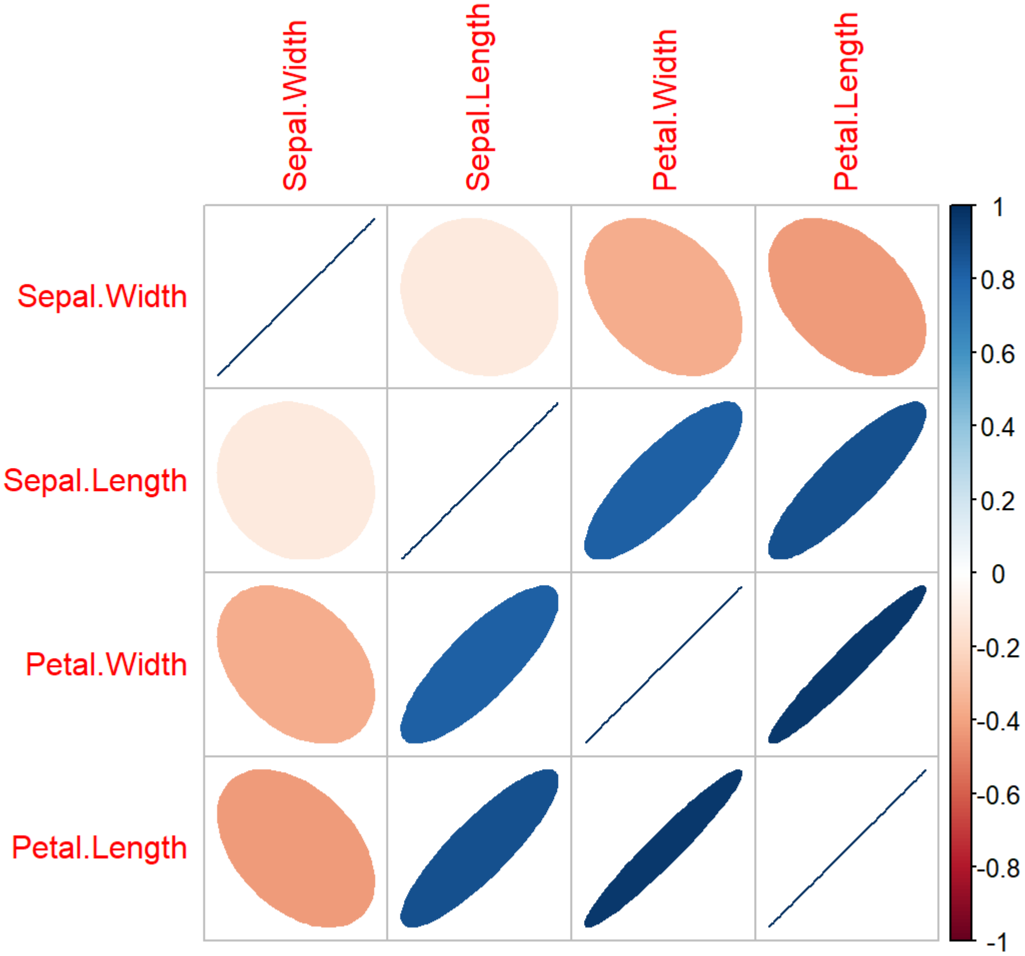
```
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```
corrplot(cm1, method="ellipse")
```



```
corrplot(cm1, method=c("ellipse"), order="FPC")
```



Test if correlation is significantly different from 0

```
cor.test(iris$Sepal.Length, iris$Sepal.Width)
```

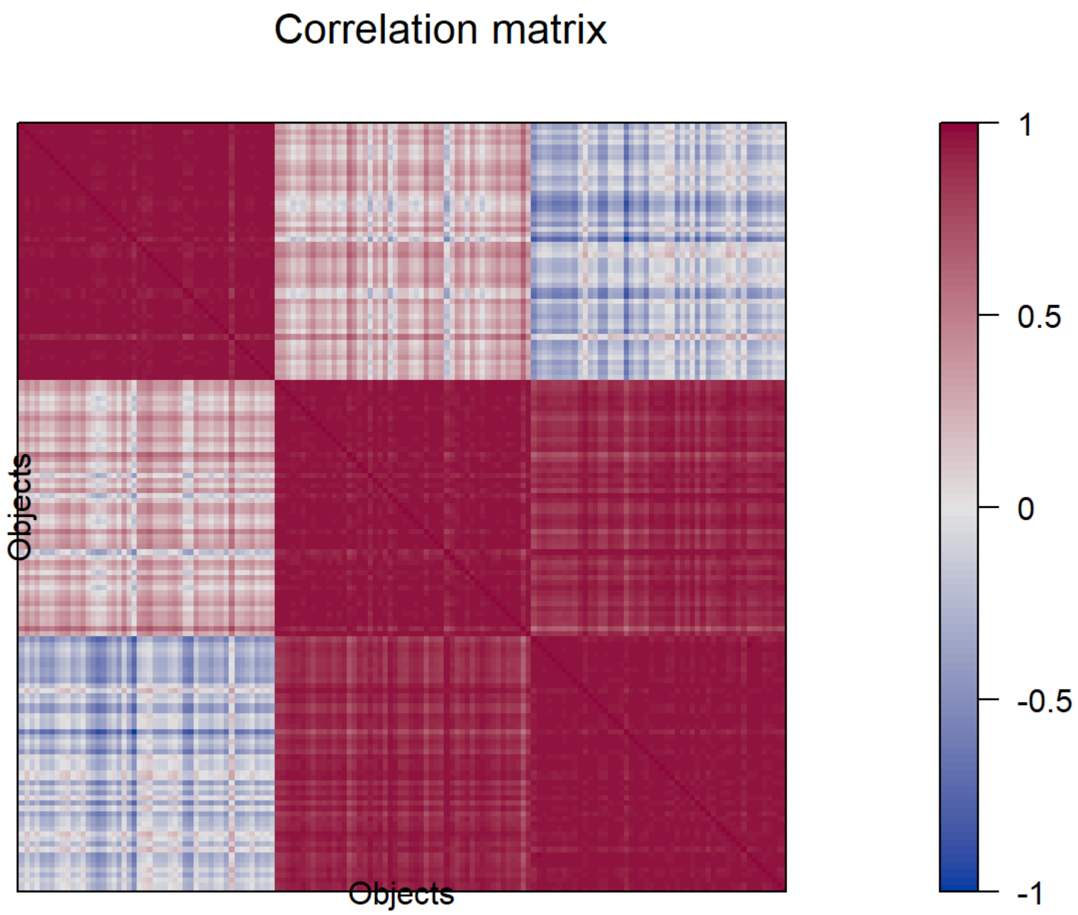
```
##
##  Pearson's product-moment correlation
##
## data:  iris$Sepal.Length and iris$Sepal.Width
## t = -1.4403, df = 148, p-value = 0.1519
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.27269325  0.04351158
## sample estimates:
##        cor
## -0.1175698
```

```
cor.test(iris$Petal.Length, iris$Petal.Width) #this one is significant
```

```
##
##  Pearson's product-moment correlation
##
## data:  iris$Petal.Length and iris$Petal.Width
## t = 43.387, df = 148, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9490525 0.9729853
## sample estimates:
##       cor
## 0.9628654
```

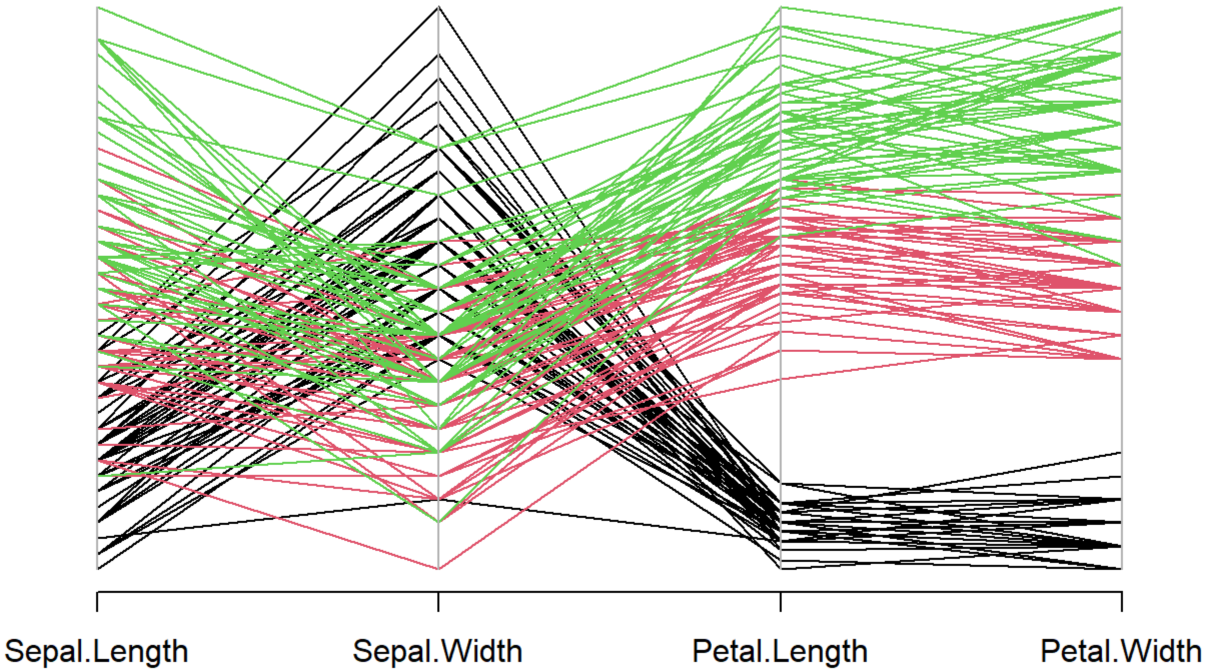
Correlation between objects

```
cm2 <- cor(t(iris_matrix))
pimage(cm2,
    main="Correlation matrix", xlab="Objects", ylab="Objects",
  zlim = c(-1,1),col = bluered(100))
```



Parallel coordinates plot

```
library(MASS)
parcoord(iris[,1:4], col=iris$Species)
```

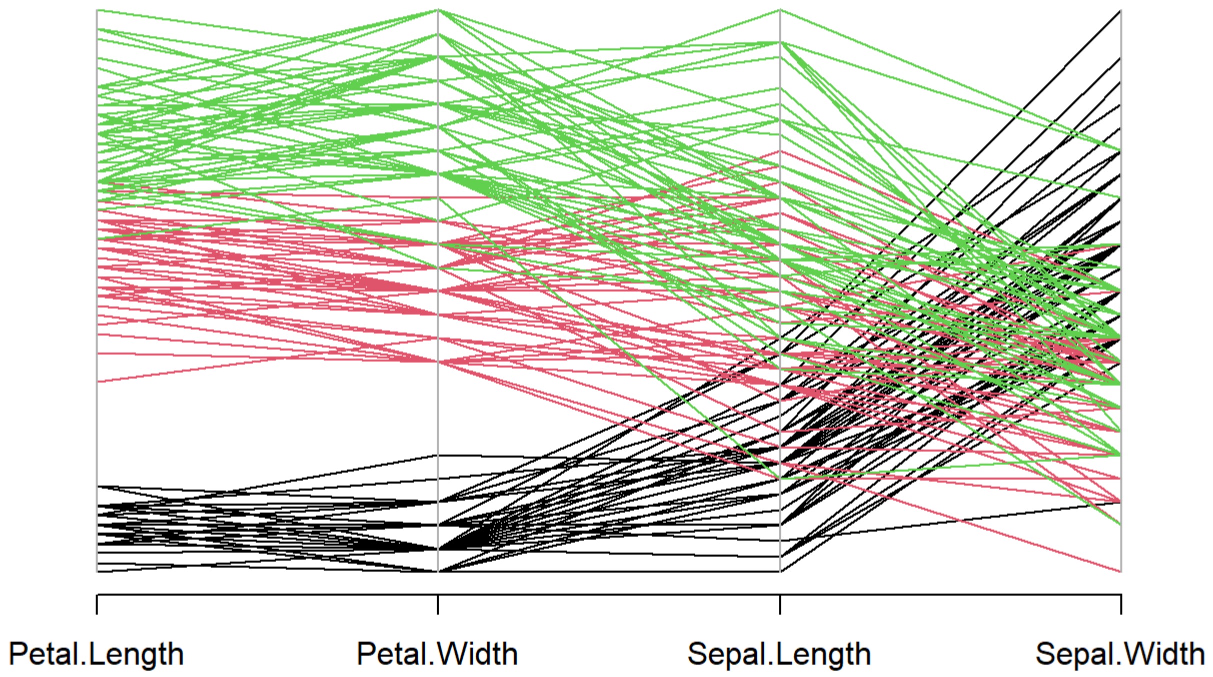


Reorder with placing correlated features next to each other

```
library(seriation)
o <- seriate(as.dist(1-cor(iris[,1:4])), method="BBURCG")
get_order(o)
```

```
## Petal.Length  Petal.Width Sepal.Length  Sepal.Width
##            3            4            1            2
```

```
parcoord(iris[,get_order(o)], col=iris$Species)
```



Look at some example maps at http://rgraphgallery.blogspot.com/search/label/map (http://rgraphgallery.blogspot.com/search/label/map)