Data Mining - Data

Dr. Jean-Michel RICHER

université angers

FACULTÉ
DES SCIENCES
Unité de formation
et de recherche
DÉPARTEMENT
INFORMATIQUE

2018

jean-michel.richer@univ-angers.fr

Outline

1. Introduction

What we will cover

Definition 1

- preprocessing of data (normalization, standardization)
- sampling to train and test classifiers
- feature selection with PCA

Raw data collection

datasets are not always in good shape

- different sources of information (Database, Excel, ...)
- missing data, errors

a first pre-processing step is needed

- for example scaling, translation, and rotation of images
- **rearrange**, remove rows with empty data
- or **imputate** values, i.e. replace missing values using certain statistics (mean, most common, ...)

2. Data preprocessing

Normalization

Normalization and other feature scaling techniques are often mandatory in order to make comparisons between different attributes if the attributes were **measured on different scales** (e.g., temperatures in Kelvin and Celsius);

- the term *normalization* is often used synonymous to *Min-Max scaling*
- The scaling of attributes in a certain range, e.g., 0 to 1:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Standardization

Another common approach is **standardization** or *scaling to unit-variance*

- every sample is subtracted by the attribute's mean μ and divided by the standard deviation σ
- attributes will have the properties of a standard normal distribution ($\mu = 0$ and $\sigma = 1$)

this is done in order not to have one property which will have more influence than the others

Sampling

split the dataset into a **training** and a **test** dataset to assess precision of method (classifier)

- training dataset is used to train the model
- test dataset evaluate the performance of the final model
- be aware of **overfitting**: classifiers that perform well on training data but do not generalize well

Cross-Validation

most useful techniques to evaluate different combinations of

- feature (property) selection (take only intersting features)
- dimensionality reduction of learning algorithms (reduce for efficiency)

multiple flavors of cross-validation, most common is **k-fold** cross-validation

K-fold cross-validation

K-fold

the original training dataset is split into k different subsets (the so-called folds)

- one fold is retained as test set
- the other $k - 1$ are used for training the model
- calculate the average error rate (and standard deviation) of the folds

use matrix of statistics to see if some fold is different from others

Feature Selection and Dimensionality Reduction

the key difference between the terms:

- feature selection: keep the original feature axis
- dimensionality reduction: usually involves a transformation technique

the main goals are:

- noise reduction
- increase computational efficiency by retaining only **useful** (discriminatory) information
- avoid overfitting

Feature Selection

retain only those features that are **meaningful** and help build a *good* classifier by reducing the number of attributes (features, properties)

- **Linear Discriminant Analysis (LDA)**: **supervised**, computes the directions (LD) that maximize the separation between multiple classes
- **Principal Component Analyses (PCA)**: **unsupervised**, ignores class labels, find PC that maximize the variance in a dataset

LDA

in a nutshell

- project a feature space (a dataset of k-dimensional samples) onto a smaller subspace h (where $h \leq k - 1$)
- while maintaining the class-discriminatory information

PCA (Principal Component Analysis

- reduce the dimensionality of a data set consisting of a large number of interrelated variable
- retain as much as possible the variation in the data set
- achieved by transforming to a new set of **uncorrelated** variables (PC)
- ordered so that the first few retain most of the variation present in all of the original variables

PCA Algorithm (1/2)

- Take the whole dataset consisting of k-dimensional samples ignoring the class labels
- Compute the k-dimensional mean vector (i.e., the means for every dimension of the whole dataset)
- Compute the scatter matrix (alternatively, the covariance matrix) of the whole data set
- Compute eigenvectors $(e_1, e_2, \ldots, e_k)$ and corresponding eigenvalues $(\lambda_1, \lambda_2, \ldots, \lambda_k)$

PCA Algorithm (2/2)

- Sort the eigenvectors by decreasing eigenvalues and choose h eigenvectors with the largest eigenvalues to form a $k \times h$ dimensional matrix W (where every column represents an eigenvector)
- Use this $k \times h$ eigenvector matrix to transform the samples onto the new subspace $y = W^T \times x$ (where x is a $k \times 1$-dimensional vector representing one sample and y is the transformed $h \times 1$-dimensional sample in the new subspace

PCA algorithm (2/2)

Types of PCA

- **general**: apply algorithm to data (no transformation of data)
- **centered**: substract mean to initial values
- **reduced**: first center then reduce (if variances are significantly different, divide each variables by its standard deviation)

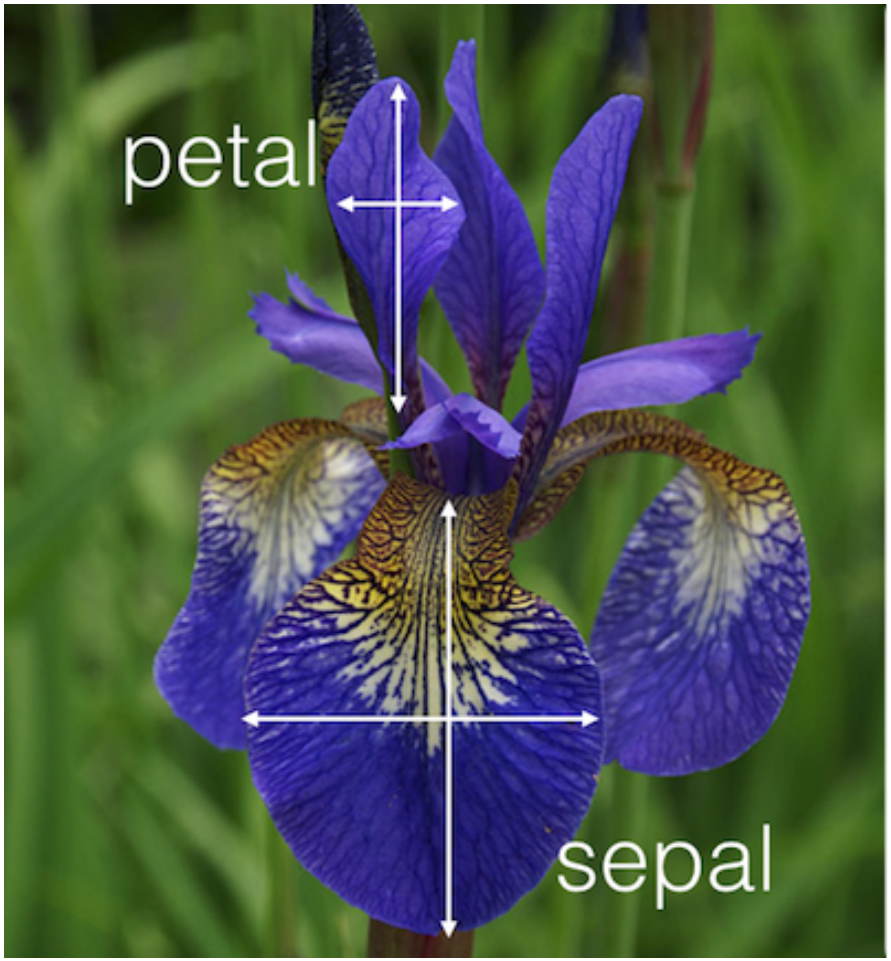
3. PCA with R

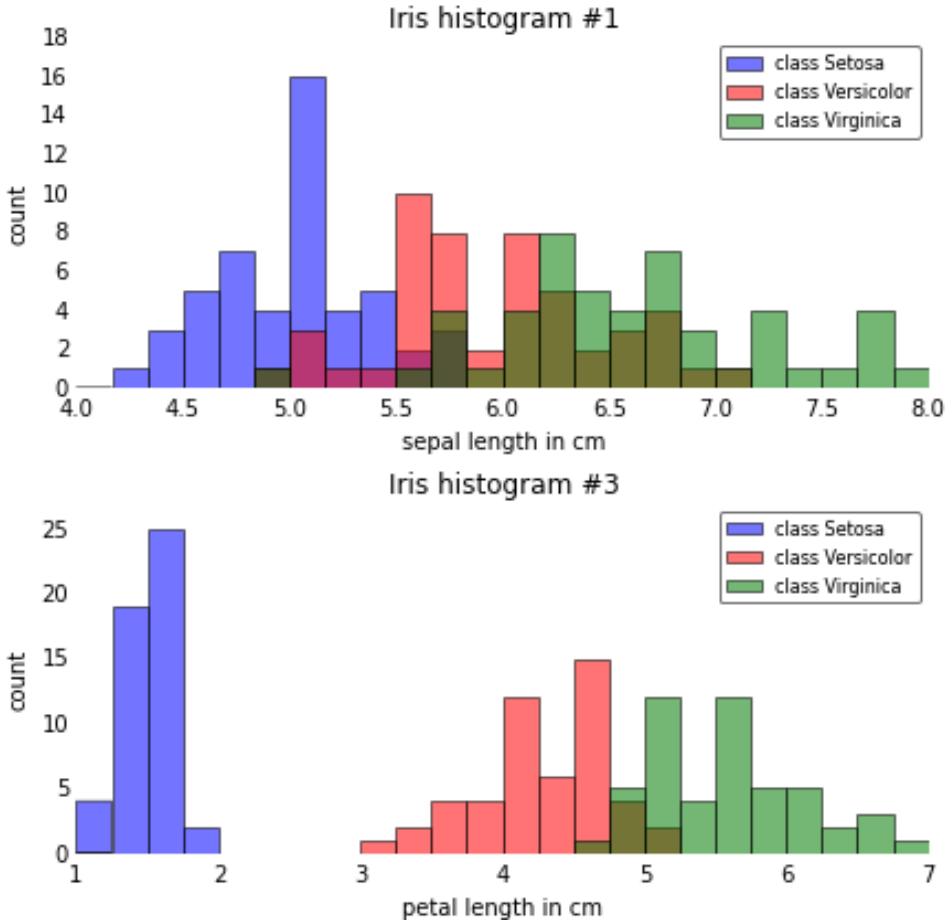
Example of PCA with R

To have a better understanding of PCA we will use the **IRIS** flowers dataset

- 150 individuals (50 Setosa, 50 Versicolour, 50 Virginica)
- 5 properties:
  - petal length and width
  - sepal length and width (protection for the flower and support for the petals when in bloom)
  - class (Setosa, Versicolour, Virginica)

Example of PCA with R





View the data

Dataset already present

```
> 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
```

Example of PCA with R

```
> 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
```

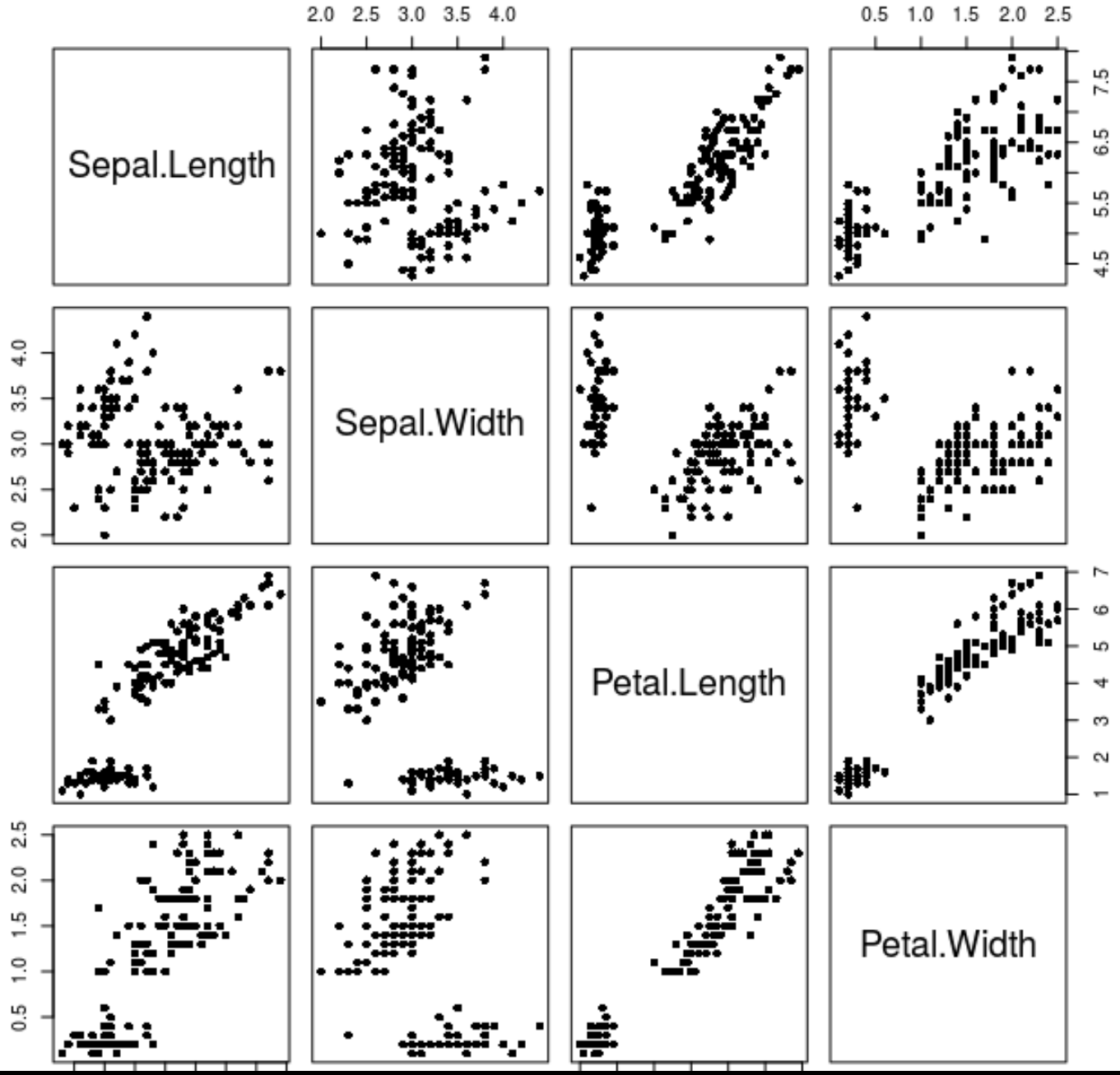
Example of PCA with R

Plot the data to have a better understanding (with hundreds of attribute this is not feasible)

```
> attach(iris)
> plot(iris[,1:4], pch=16)
```

what can we really see ?

Example of PCA with R

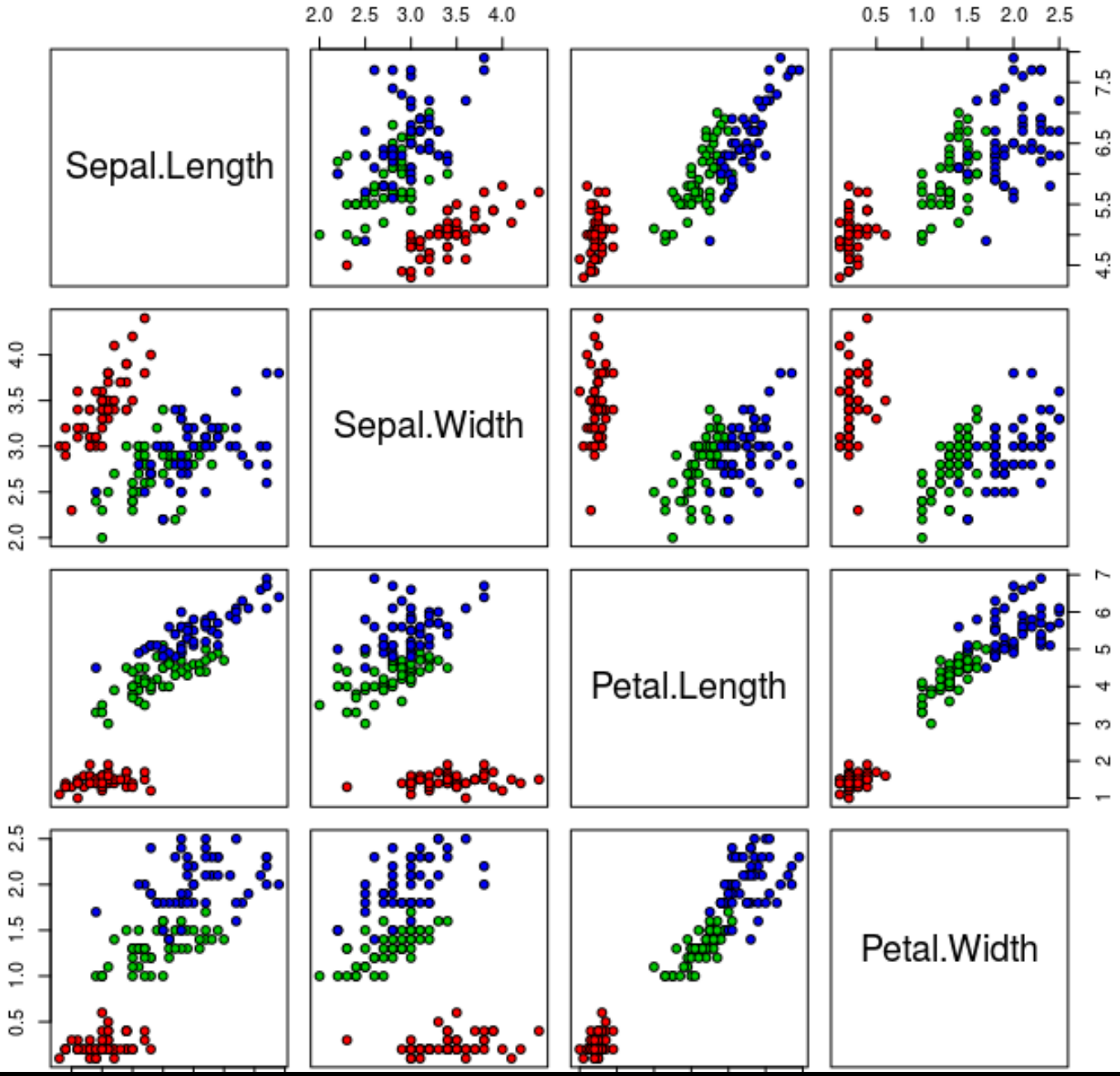


Example of PCA with R

Plot in function of classes:

```
> attach(iris)
> pairs(iris[1:4],pch = 21, bg = c("red", "green3",
  "blue")[unclass(iris$Species)])
```

Example of PCA with R



Example of PCA with R

In this case

- if you compare *Sepal.Length* to *Sepal.Width*:
  - the red dots are separated from the blue and green dots
  - but it will be difficult to differentiate blue from green
- if you compare *Petal.Length* to *Petal.Width* the differentiation is simple

Example of PCA with R

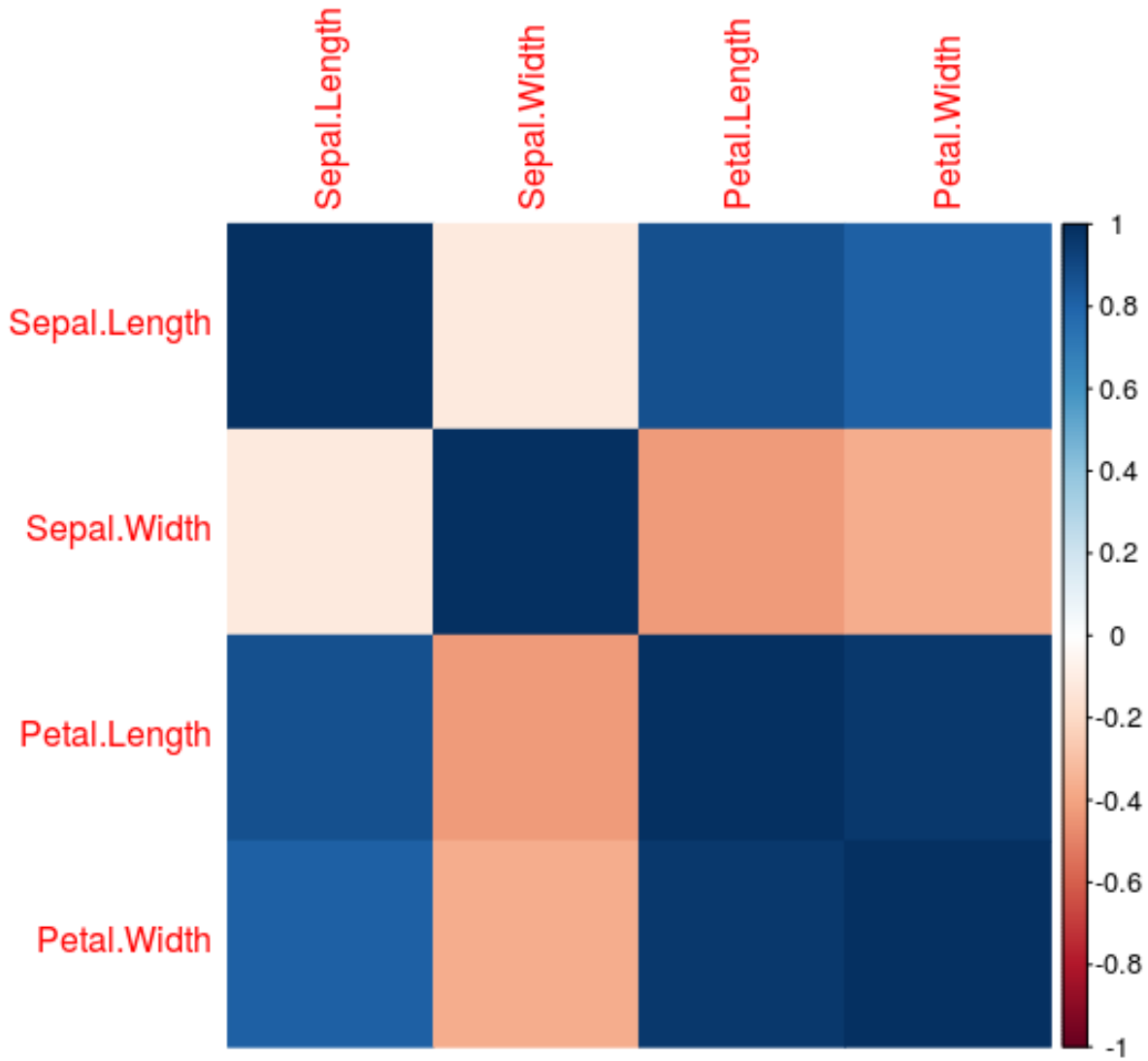
Correlation matrix values

```
> cor(iris[,1:4])
             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
```

Then plot the correlation matrix

```
library(corrplot)
corrplot(cor(iris[,1:4]), method="color")
```

Example of PCA with R



Correlation explanation

if the correlation value is

- close to 1, then both vectors of values are strongly correlated
- close to 0, then they are not correlated

In this example *Sepal.Length* and *Petal.Length* are strongly correlated, so it is interesting to remove one of them

Example of PCA with R

What we want to know

- is there a difference between the species ?
- which properties can explain thoses differences ?

Example of PCA with R

Perform the PCA with R

```
pca = prcomp(iris[,1:4])
```

The principal component are obtained with `pca$rotation`

Principal components

```
> pca$rotation
                    PC1         PC2         PC3         PC4
Sepal.Length  0.36138659 -0.65658877  0.58202985  0.3154872
Sepal.Width  -0.08452251 -0.73016143 -0.59791083 -0.3197231
Petal.Length  0.85667061  0.17337266 -0.07623608 -0.4798390
Petal.Width   0.35828920  0.07548102 -0.54583143  0.7536574
```

$$PC1 = \begin{cases} & 0.86 & \times & \textit{Petal.Length} \\ + & 0.36 & \times & \textit{Sepal.Length} \\ + & 0.36 & \times & \textit{Petal.Width} \\ - & 0.08 & \times & \textit{Sepal.Width} \end{cases}$$

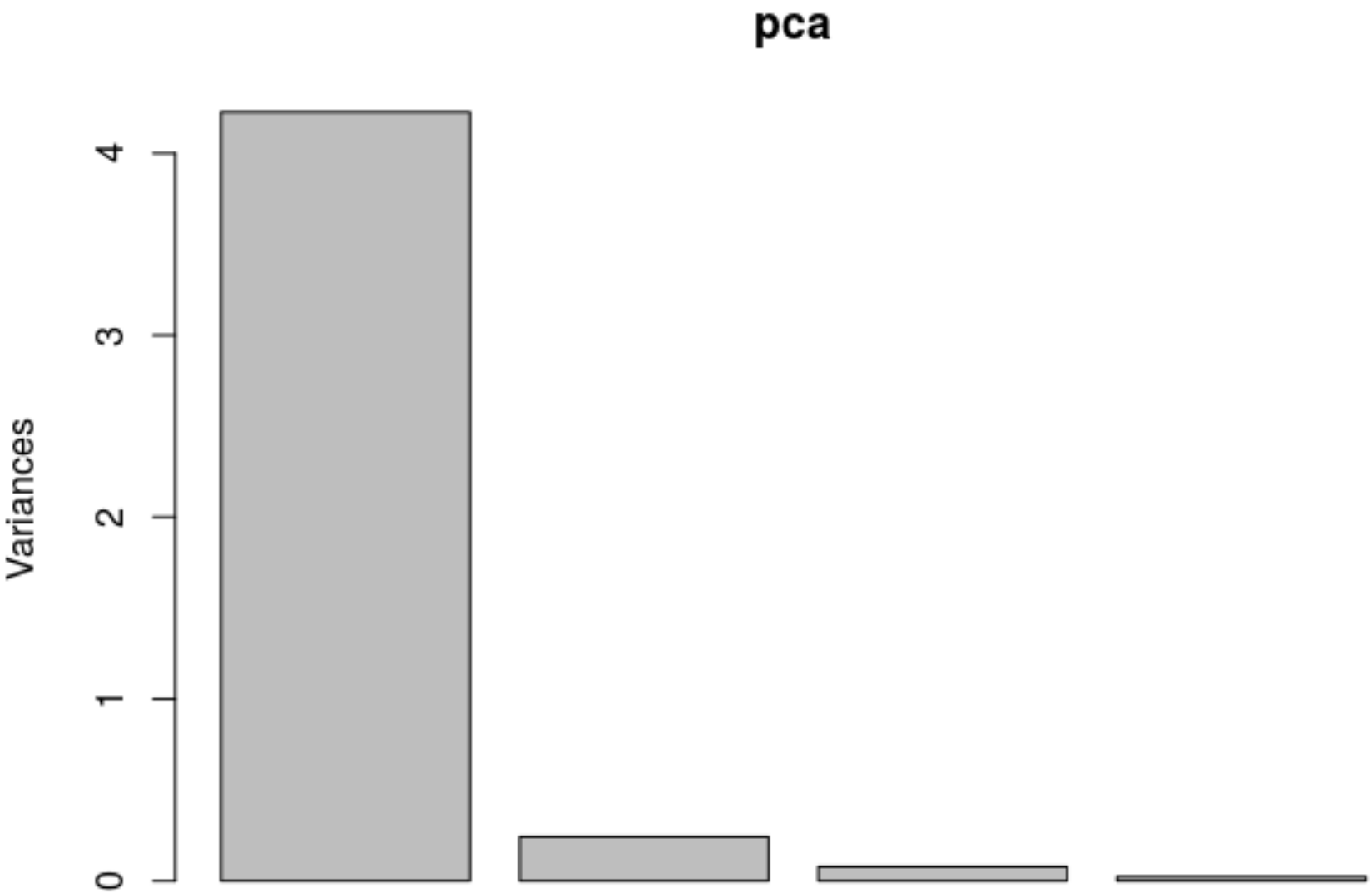
Principal components information

`pca$sdev` gives the pourcentage of information in each component:

```
> pca$sdev
[1] 2.0562689 0.4926162 0.2796596 0.1543862
> 100 * pca$sdev^2 / sum(pca$sdev^2)
[1] 92.4618723  5.3066483  1.7102610  0.5212184
> sum(100 * (pca$sdev^2)[1:1] / sum(pca$sdev^2))
[1] 92.46187
> sum(100 * (pca$sdev^2)[1:2] / sum(pca$sdev^2))
[1] 97.76852
> sum(100 * (pca$sdev^2)[1:3] / sum(pca$sdev^2))
[1] 99.47878
> pca$rotation
```

Already 92% of the information is given by the first component !

Example of PCA with R



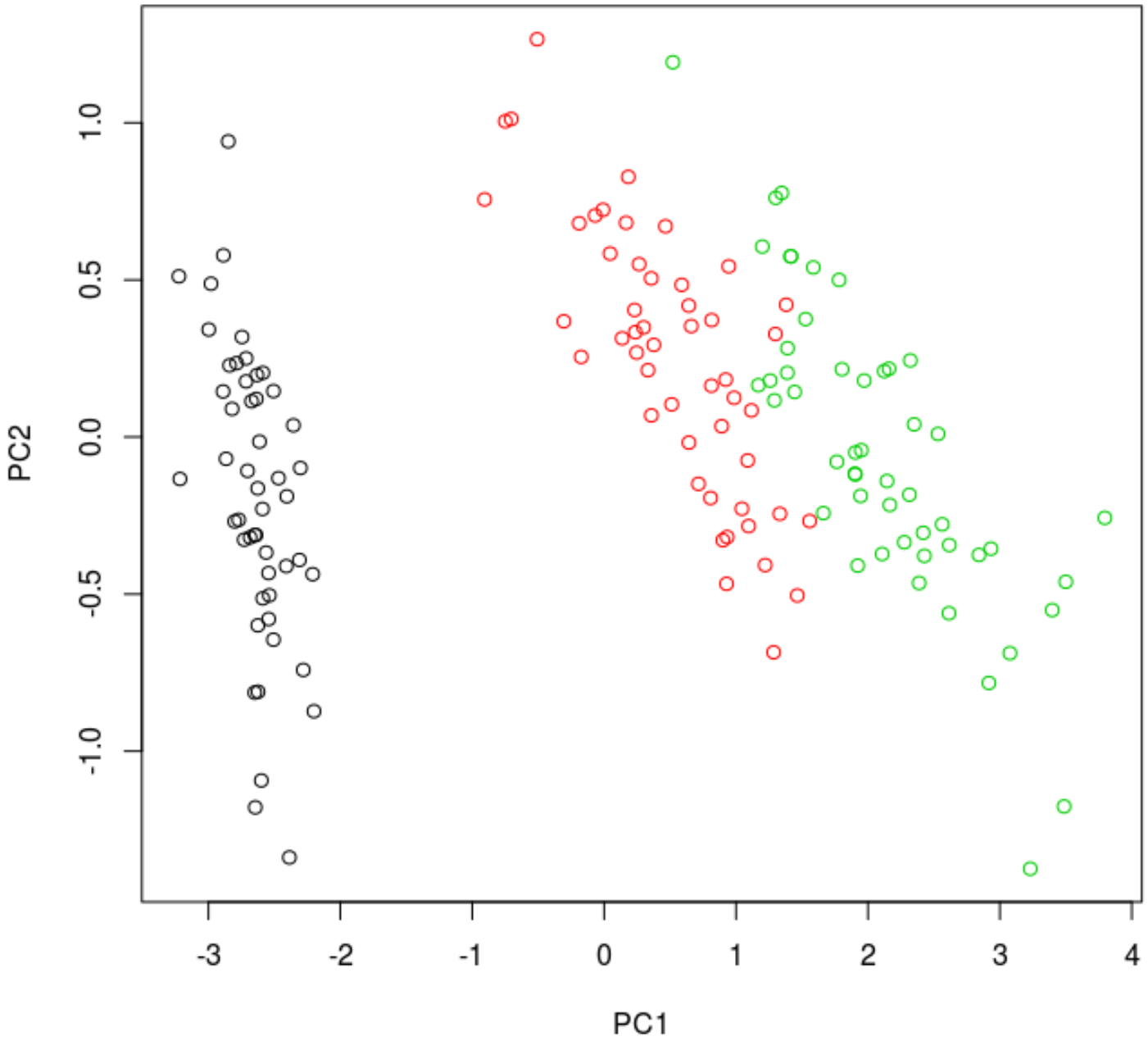
Print individuals in function of PC

```
> plot(pca$x[,1], rep(0,nrow(iris)), col=iris[,5])
```

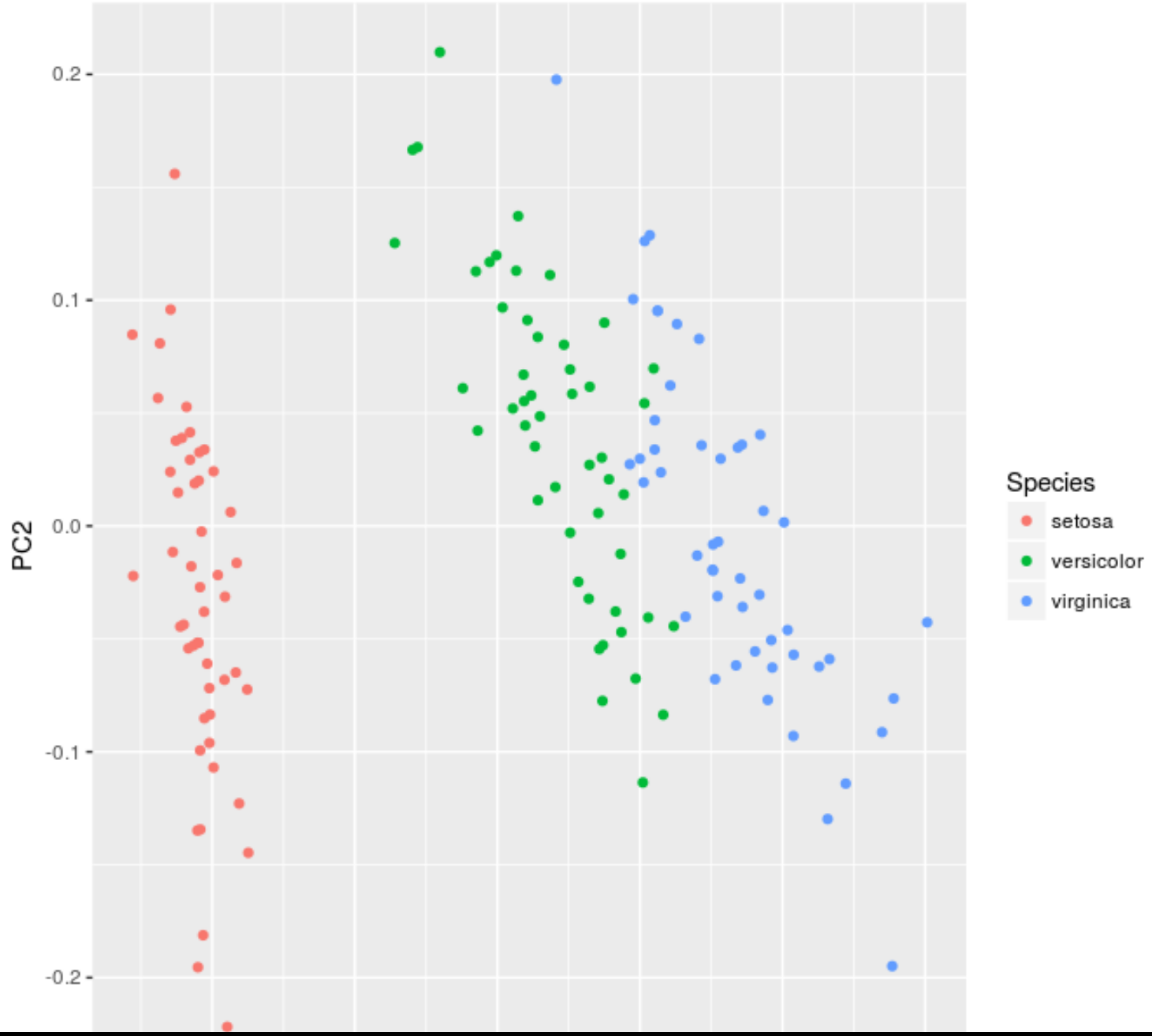
or

```
> library(ggfortify)
> df <- iris[c(1, 2, 3, 4)]
> autoplot(prcomp(df), data = iris, colour = 'Species')
> autoplot(prcomp(df), data = iris, colour = 'Species',
  label = TRUE, label.size = 3)
```

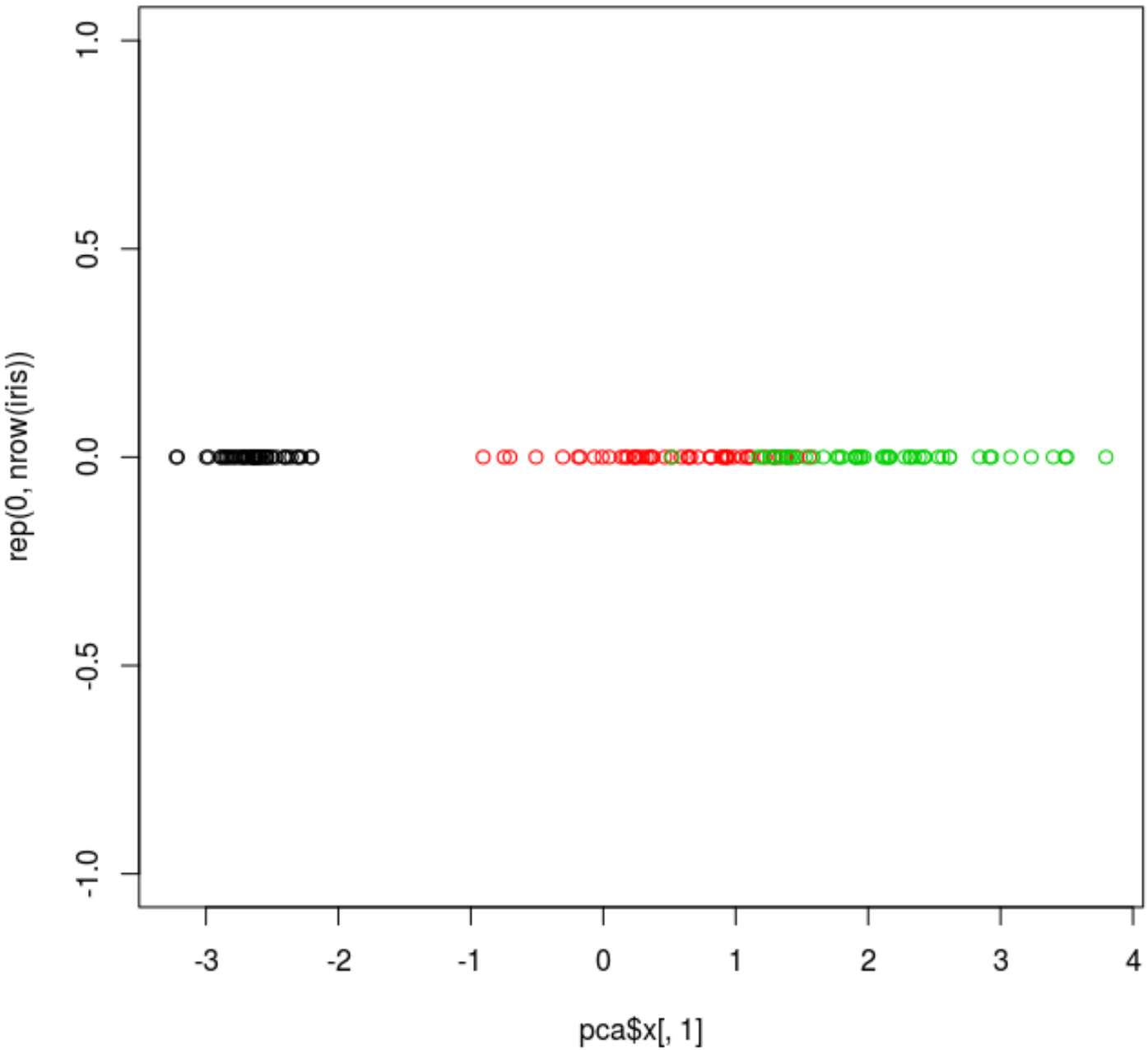
Example of PCA with R



Example of PCA with R



Example of PCA with R



Example of PCA with R

Final interpretation

- setosa (red) can be identified clearly
- as the PC 1 is composed of $0.86 \times$ *Petal.Length*, it is the length of the petal that can help differentiate species

More information about PCA

More information

Follow the following links

- http://www.sthda.com/english/wiki/print.php?id=206
- https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/

4. Exercise and home work

Exercise about PCA

PCA with python

install the sklearn package and try to do as R

- display information: data, correlation matrix
- perform the PCA, print results
- try to plot data

Home work about PCA

PCA with Weka

- write a report explaining how to perform PCA using Weka
- give detail of each step

4. End

FACULTÉ
DES SCIENCES
Unité de formation
et de recherche
DÉPARTEMENT
INFORMATIQUE

UA - Angers
2 Boulevard Lavoisier 49045 Angers
Cedex 01
Tel: (+33) (0)2-41-73-50-72

université
angers
FACULTÉ
DES SCIENCES
Unité de formation
et de recherche
DÉPARTEMENT
INFORMATIQUE