GRAU EN ENGINYERIA DE DADES

104365 Visualització de Dades

Teoria 6. Tractament de dades II

Departament de Matemàtiques

Escola d'Enginyeria UAB

UAB

Data processing for visualization

- **Chapter 5 - Data processing for visualization (I)**
  - Uncertainty and error
  - Transformations and data massage (+ seminars & PRT1)
- **Chapter 6 (today) - Data processing for visualization (II)**
  - Dimensionality reduction
  - Computation and important metrics selection

6. Data processing for visualization (II). Contents:

1. Dimensionality reduction

1. Introduction
2. Correlograms
3. Feature projection – PCA
4. Discriminant analysis (linear) - LDA
5. T-Distributed stochastic neighbour embedding (t-SNE)
6. Tomography- Slice along a plane, 2D isosurfaces for a 3D field, isocontours

2.Computation and important metrics selection

1. Quality metrics: Noise reduction, clutter reduction, search outliers

6.1.1. Introduction: **Dimensionality reduction**

Dimensionality reduction (DR) means: the process of **transformation of data from high dimensional space to low dimensional space** while maintaining most of the meaningful insights from the original data.

The goal is to preserve the meaningful structure of a dataset while using fewer attributes to represent the items.

For example: We have a dataset contains hundred columns (i.e features) or it could be an array of points that make up a large sphere in the 3D space. DR?

DR entails lowering the number of columns to a smaller number, such as 2D.

6.1.1. Introduction: **Dimensionality reduction**

Dimensionality reduction (DR) has two primary use cases:

– data exploration

– machine learning.

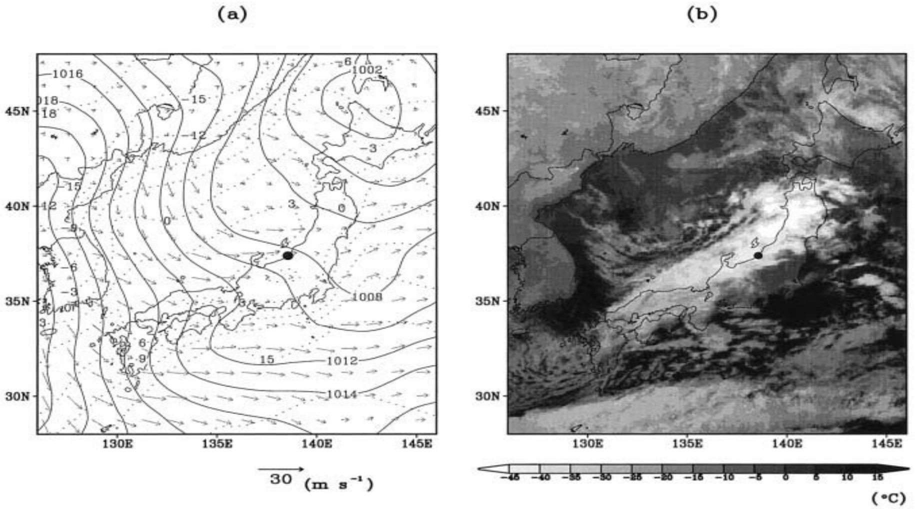
DR is a strategy for managing complexity in visualization:

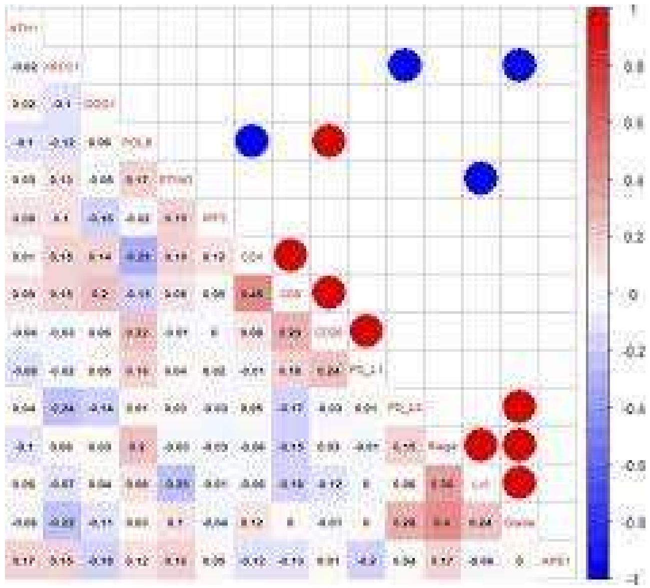
It is useful *for data exploration* because **dimensionality reduction to few dimensions** (e.g., 2D or 3D) **allows for visualizing the samples.**

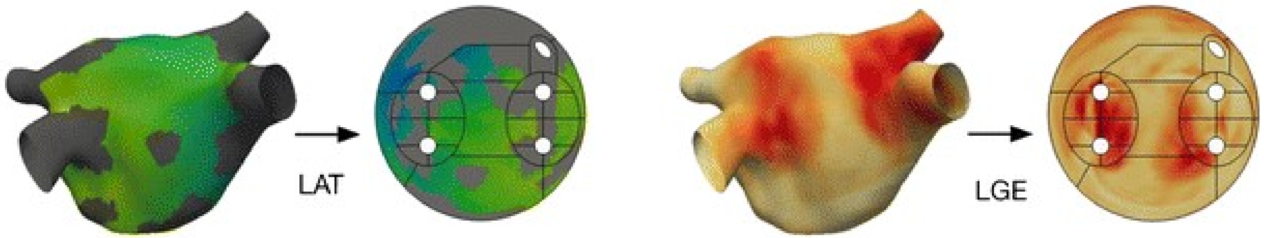
Such a visualization can then be used to obtain insights from the data *(e.g., detect clusters and identify outliers).*

6.1.1. Introduction: **Dimensionality reduction**

Examples of dimension reduction:

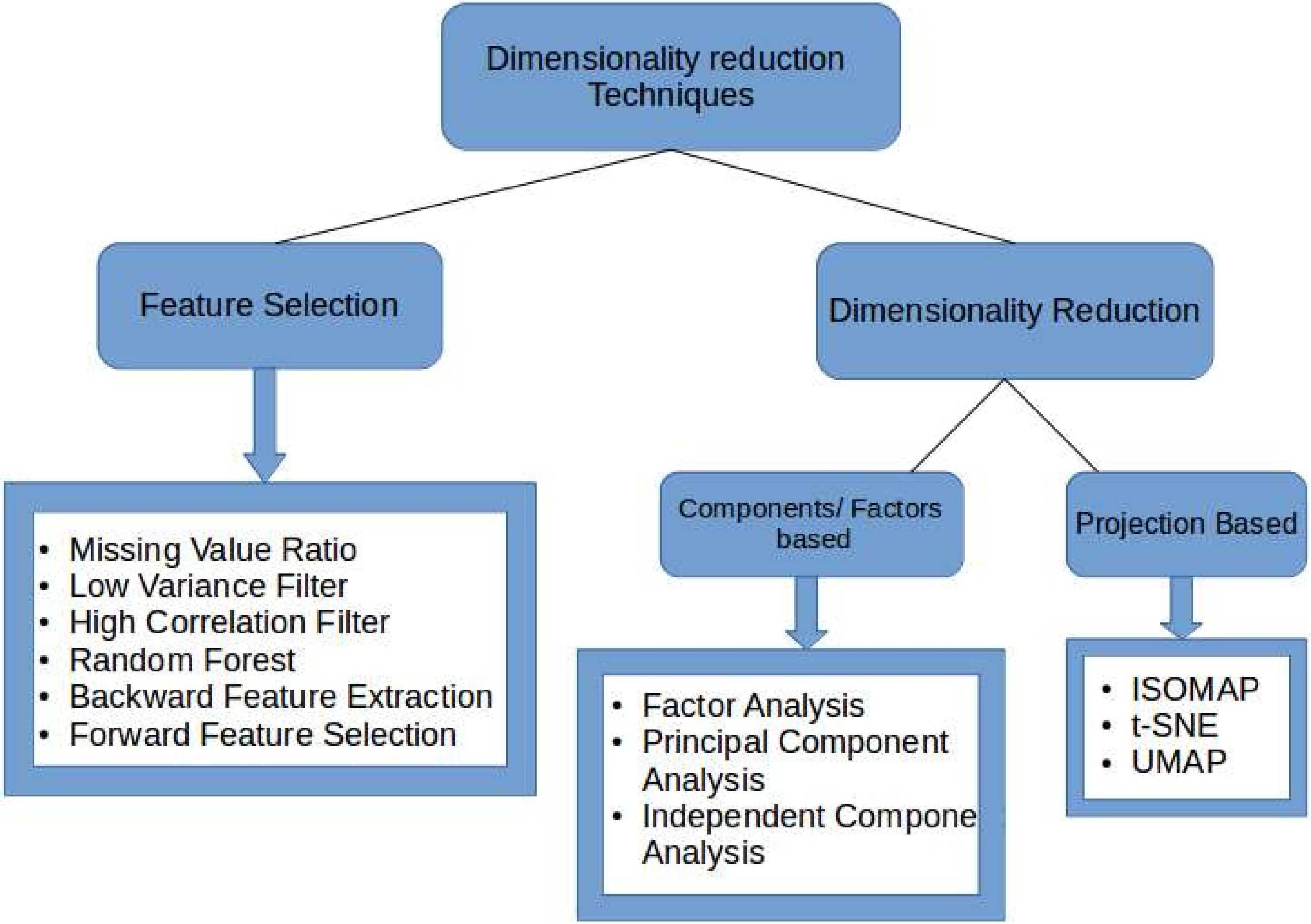






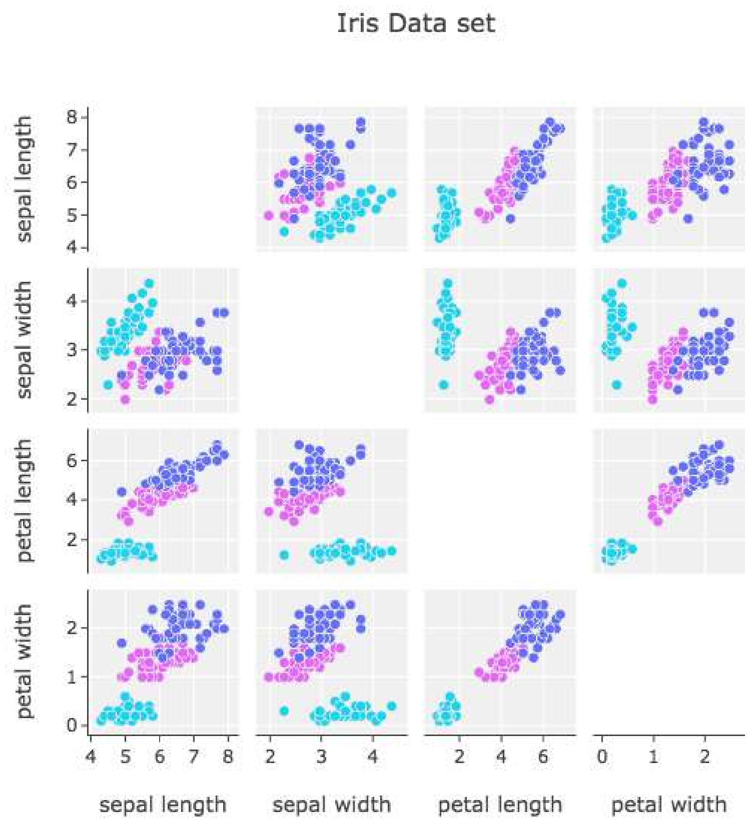
DOI: 10.1007/s10840-017-0281-3

6.1.1. Introduction: **Dimensionality reduction**



6.1.1. Remember: **Scatterplot matrix limitation**

We saw: **Scatterplot matrix (SPLOM)** uses multiple scatterplots *to determine the correlation (if any) between a series of variables*.



!! When we have >3 or 4 quantitative variables – scatterplot matrices quickly become unwieldy

6.1.1. Correlation coefficients

We saw: **Scatterplot matrix (SPLOM)** uses multiple scatterplots to determine *the correlation (if any) between a series of variables.*

!! When we have >3 or 4 quantitative variables – scatterplot matrices quickly become unwieldy

In this case, it is **more useful to quantify the amount of association between pairs of variables and visualize these quantities rather than the raw data**.
One common way to do this is to calculate **correlation coefficients.**

6.1.1. Correlation coefficient

- Having two sets of observations: x_i and y_i
- And: $\bar{x}$ and $\bar{y}$ the corresponding sample means

The correlation coefficient is:

$$R = \frac{\sum_i (x_i - \bar{x})(yi - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y - \bar{y})^2}}$$

The **correlation coefficient *R*** is **a number between -1 and 1 that measures to what extent two variables are correlated**

6.1.1. Correlation coefficients

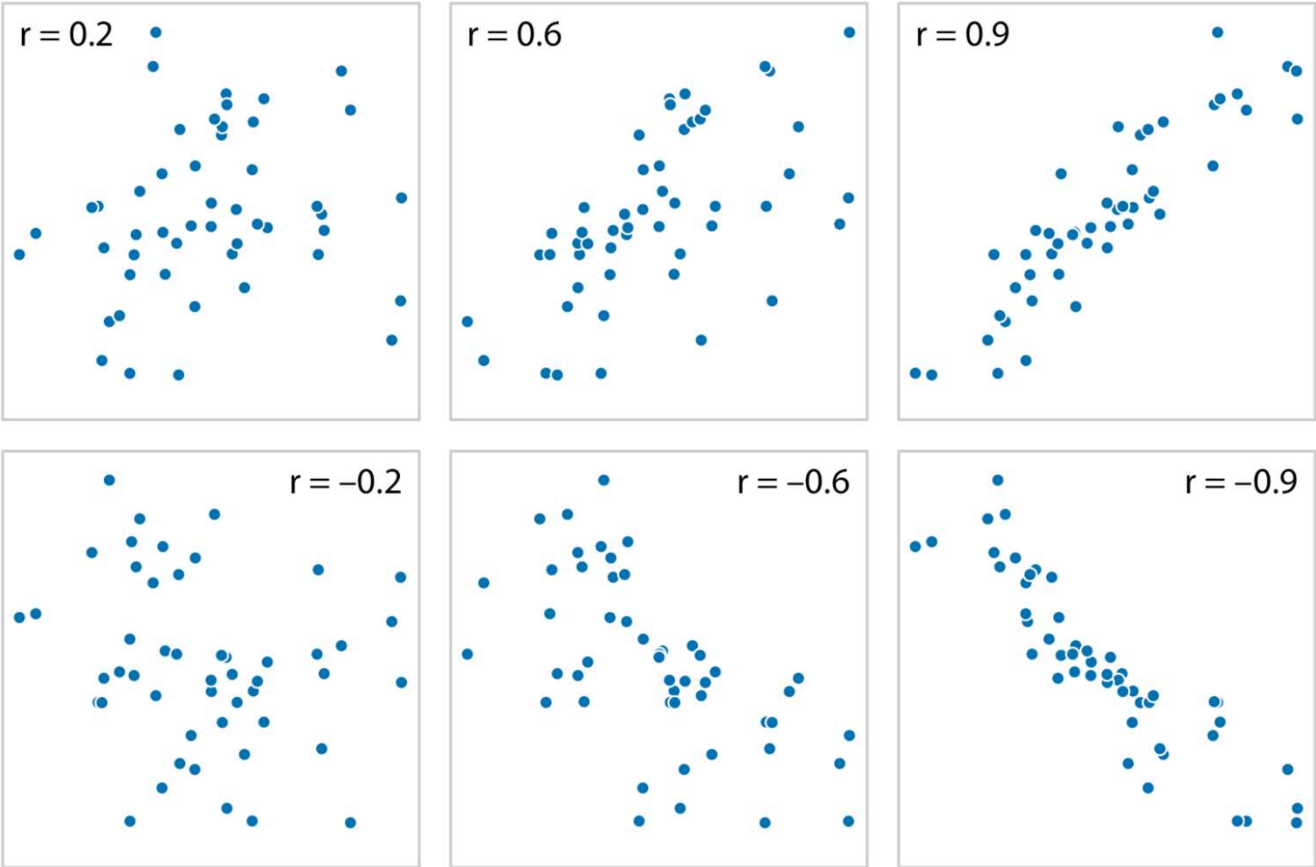
$$R = \frac{\sum_i (x_i - \bar{x})(yi - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y - \bar{y})^2}},\ -1 < R < 1$$

- R **= *0*** means there is **no association** whatsoever
- R **= *1*** *or* ***-1*** indicates a **perfect association**

The sign of the correlation coefficient R indicates :

- R **>0**: variables are **correlated** (larger values in one variable coincide with larger values in the other)
- R **<0**: **anticorrelated** (larger values in one variable coincide with smaller values in the other)

6.1.1. Correlation coefficients



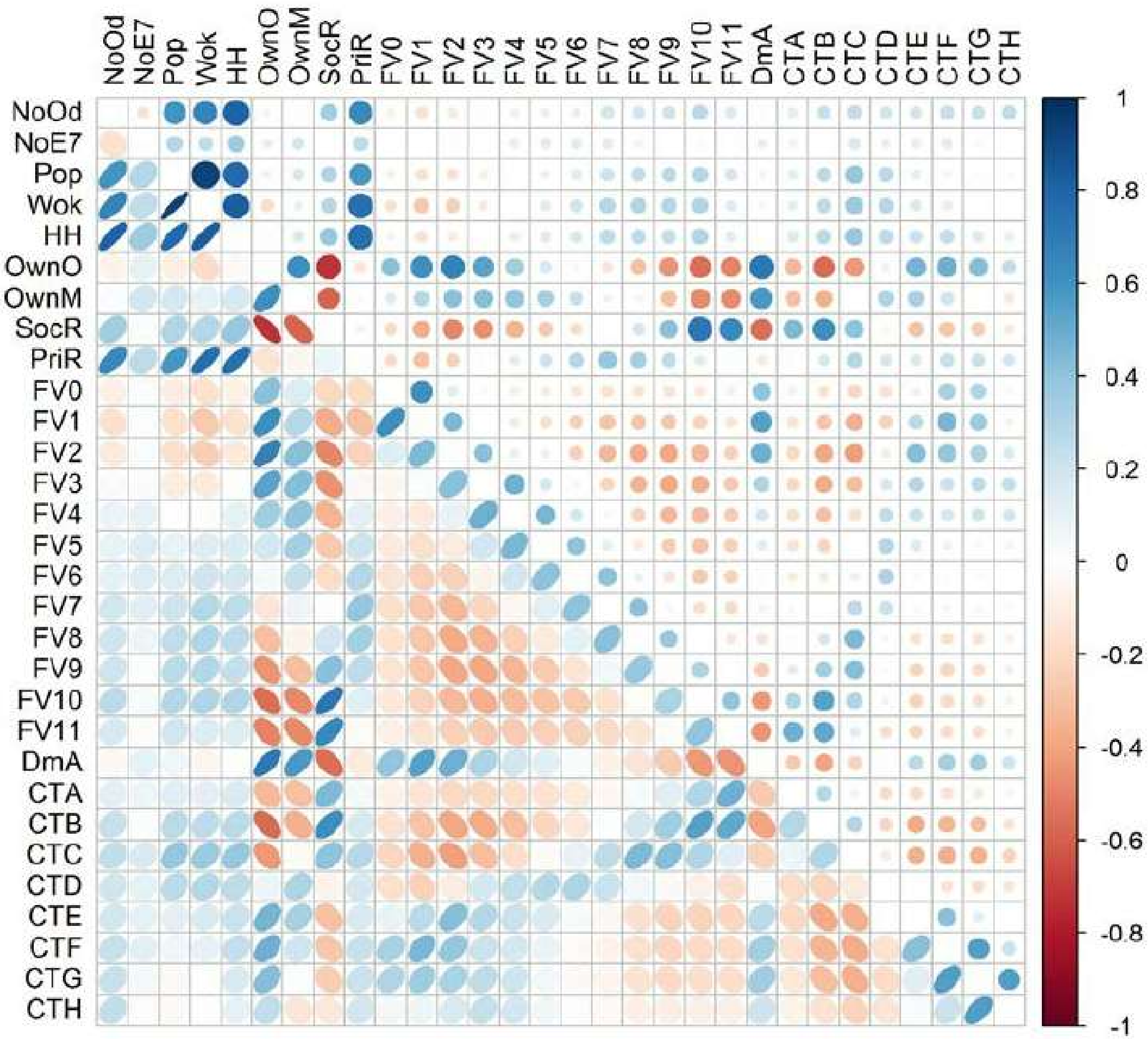
R>0

R<0

Wilke

Examples of correlations of different magnitude and direction, with associated correlation coefficient R

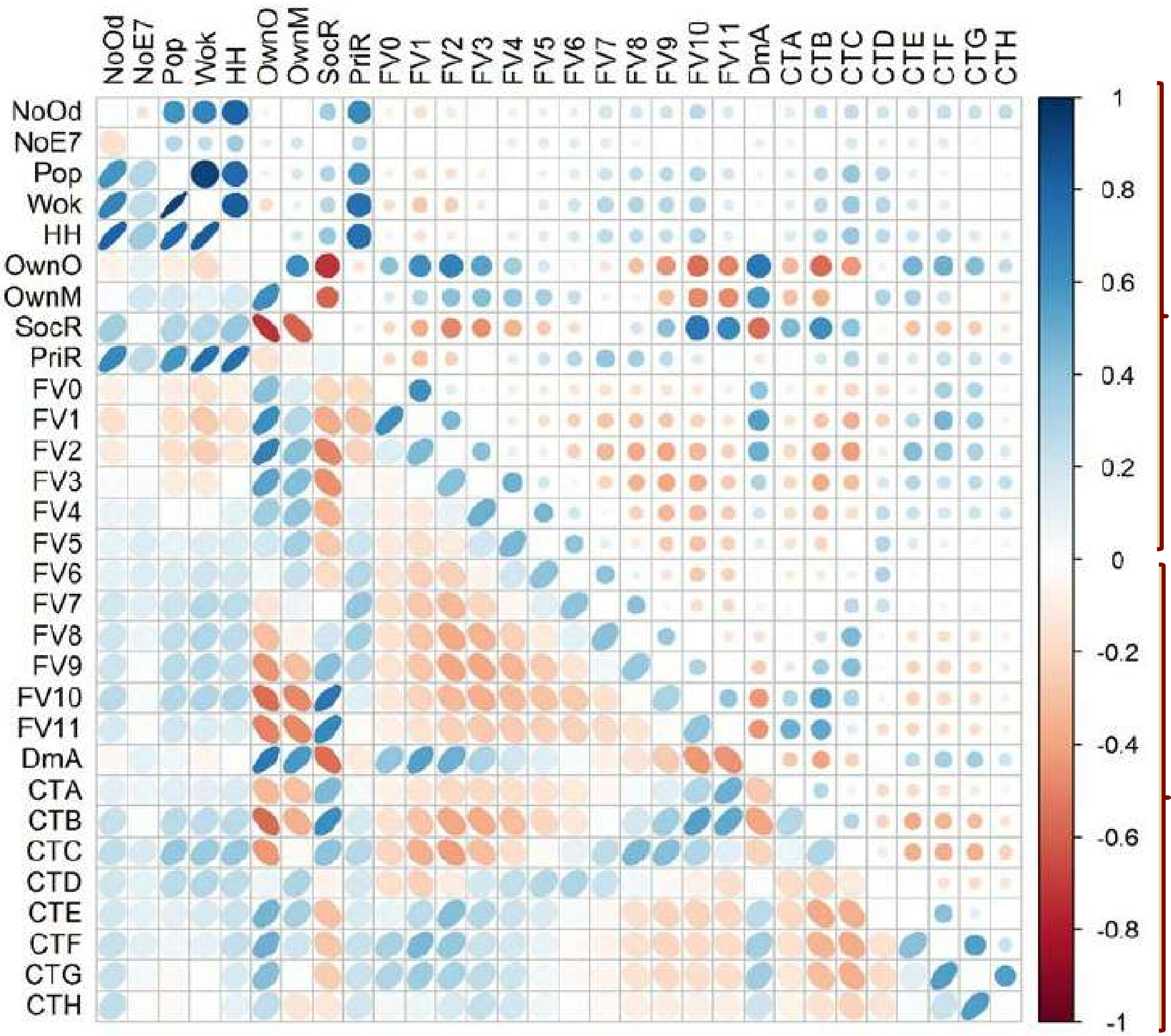
6.1.2. Correlogram



Correlation structure of the input variables. **Colour intensity and ellipse shape** are directly linked to correlation coefficient

DOI: 10.1016/j.proeng.2015.08.1069

6.1.2. Correlogram

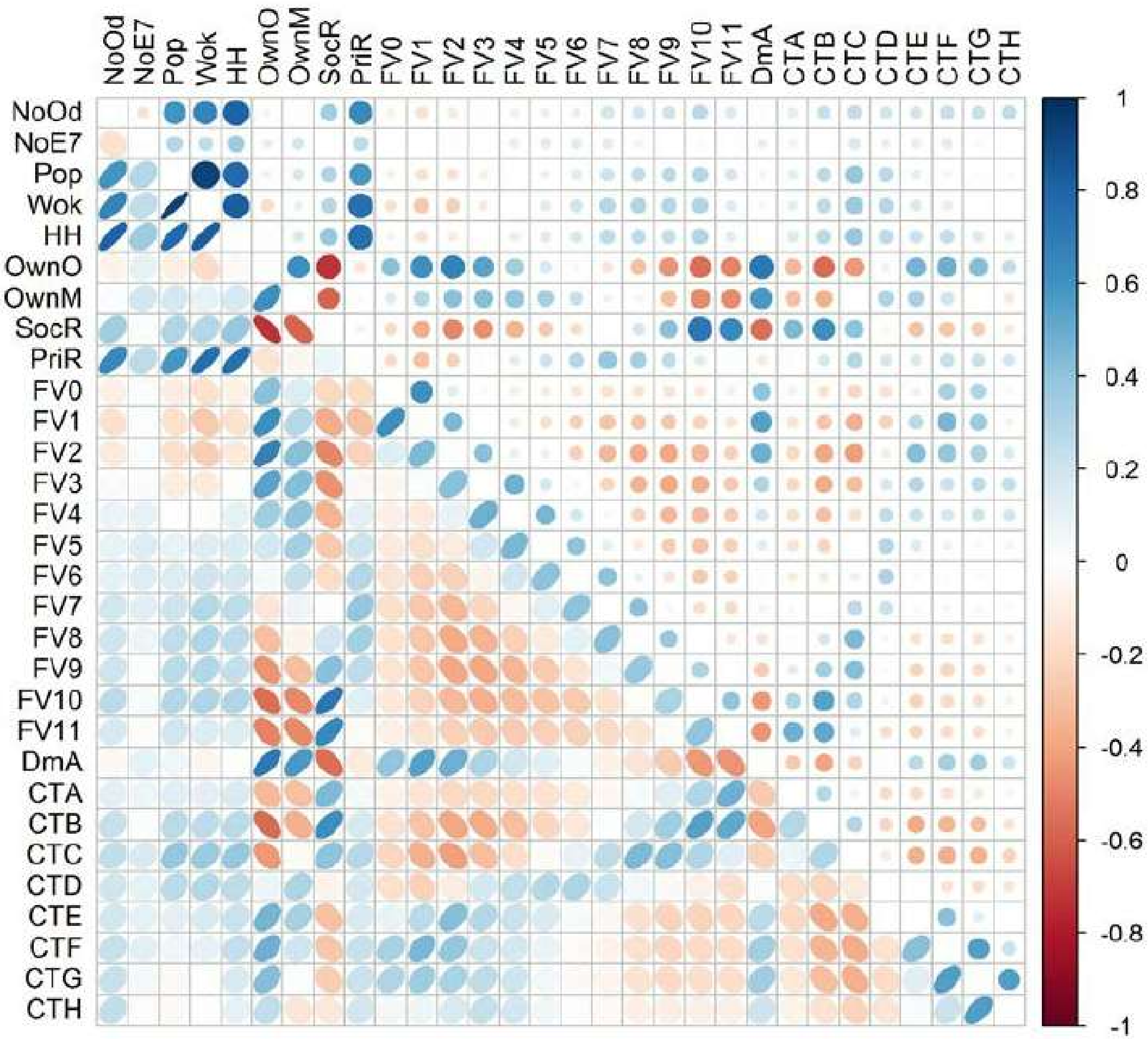


Correlation structure of the input variables. Colour intensity and ellipse shape are directly linked to correlation coefficient

R>0 is displayed in blue (correlated)

R<0 is shown in red (anticorrelated).

6.1.2 Correlogram



- : there is a **higher negative correlation** coefficient between two variables.
- : indicates a **weak correlation** for two factors
- : there is a **higher positive correlation** coefficient between two variables

6.1. Correlation & Dimensionality reduction

DR relies on the key insight that **most high-dimensional datasets consist of multiple correlated variables** that convey overlapping information

Such datasets can be reduced to a smaller number of key dimensions without loss of much critical information.

DR can be achieved by:

- ***Feature elimination*** – we reduce the feature space by elimination feature
- ***Feature selection*** – process of selecting required features from all the features available in data. *Goal: to choose features that represent the dataset perfectly*
- ***Feature Engineering*** – process of **transforming raw data into feature**, which represent the dataset well

6.1. Correlation & Dimensionality reduction

DR relies on the key insight that **most high-dimensional datasets consist of multiple correlated variables that convey overlapping information**

Such datasets can be reduced to a smaller number of key dimensions without loss of much critical information.

There are many techniques for dimension reduction. We will see:

- Principal Components Analysis (PCA)
- Linear Discriminant Analysis (LDA)
- T-Distributed Stochastic Neighbour Embedding (t-SNE) (next day)

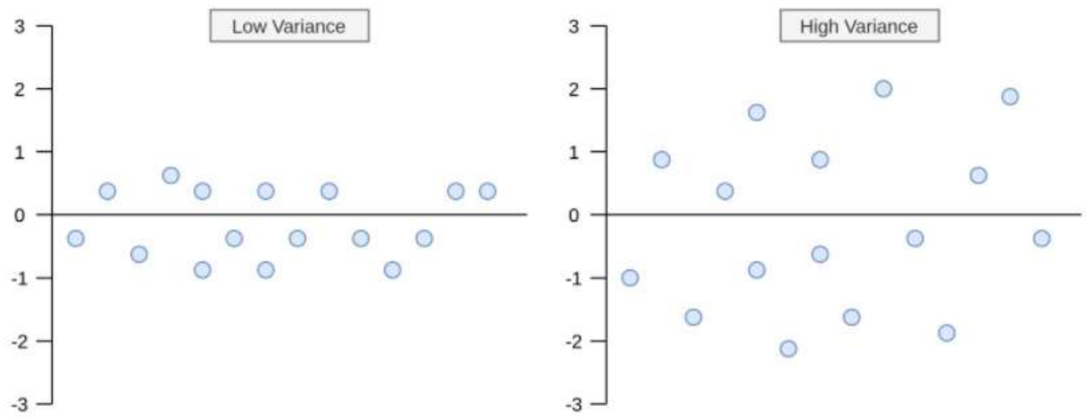
6.1. Remember: **Variance**

Before going further, **let's clarify some concepts:**

- We know that **variance** represents **the variation of values in a single variable.** *It depends on* ***how the values far from each other.***

Having a set of observations: x_i and $\bar{x}$ the corresponding sample mean:

$$var(x) = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}$$



Sergen Cansiz

6.1. Remember: **Covariance**

Before going further, let's clarify some concepts:

- Unlike the variance, **covariance** is calculated **between two different variables**. Its purpose is to find the value that indicates **how these two variables vary together**.

- Having two sets of observations: x_i and y_i
- And: $\bar{x}$ and $\bar{y}$ the corresponding sample means

$$cov(x,y) = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{N-1}$$

6.1. Remember: **Covariance vs correlation**

Before going further, let's clarify some concepts:

- **Covariance vs correlation**

- Having two sets of observations: x_i and y_i
- And: $\bar{x}$ and $\bar{y}$ the corresponding sample means

$$cov(x,y) = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{N-1}$$

$$var(x) = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N-1}$$

$$R(x,y) = \frac{\sum_i (x_i - \bar{x})(yi - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y - \bar{y})^2}}$$

6.1. Covariance vs correlation

Before going further, let's clarify some concepts:

- **Covariance vs correlation**

Covariance	Correlation
Covariance is a **measure** to indicate the extent to which **two random variables change *in tandem***	Correlation is a **measure** used to represent how strongly **two random variables are related *to each other***
Covariance **indicates the *direction* of the linear relationship** between variables	Correlation **measures *both the strength and direction* of the linear relationship** between variables
Covariance can **vary *between*** $-\infty$ ***and*** $+\infty$	Correlation **ranges *between -1 and 1***

6.1. Covariance vs correlation

Before going further, let's clarify some concepts:

- **Covariance vs correlation**

Covariance	Correlation
Covariance is affected by the change in scale. If all the values of one variable are multiplied by a constant and all the values of another variable are multiplied, by a similar or different constant, then the covariance is changed	**Correlation is NOT influenced by the change in scale**
Covariance of two *dependent variables* measures **how much in real quantity** (i.e, e.g., cm, km, liters) **on average they *co-vary.***	Correlation of two *dependent variables* measures **the proportion of how much on average these variables *vary with respect to one another***
Covariance is **zero for independent variables**	Completely **independent variables have a zero correlation**

6.1. Covariance matrix

Before going further, let's clarify some concepts:

- **Covariance matrix**

Because covariance can only be calculated between two variables, **covariance matrices** stand for representing **covariance values of each pair of variables in multivariate data.** Also, the covariance between the same variables equals variance, so, **the diagonal shows the variance of each variable**

Symmetric matrix

$$\begin{array}{cc} & \begin{array}{cc} x & y \end{array} \\ \begin{array}{c} x \\ y \end{array} & \begin{bmatrix} var(x) & cov(x,y) \\ cov(x,y) & var(y) \end{bmatrix} \end{array} \qquad \begin{array}{cc} & \begin{array}{ccc} x & y & z \end{array} \\ \begin{array}{c} x \\ y \\ z \end{array} & \begin{bmatrix} var(x) & cov(x,y) & cov(x,z) \\ cov(x,y) & var(y) & cov(y,z) \\ cov(x,z) & cov(y,z) & var(z) \end{bmatrix} \end{array}$$

2 and 3- dimensional covariance matrices

6.1. Covariance matrix

Before going further, let's clarify some concepts:

- **Covariance matrix**

These *values* in the covariance matrix **show the distribution magnitude and direction** of multivariate data in multidimensional space.

By controlling these values we can have *information about how data spread among two dimensions.*

Symmetric matrix

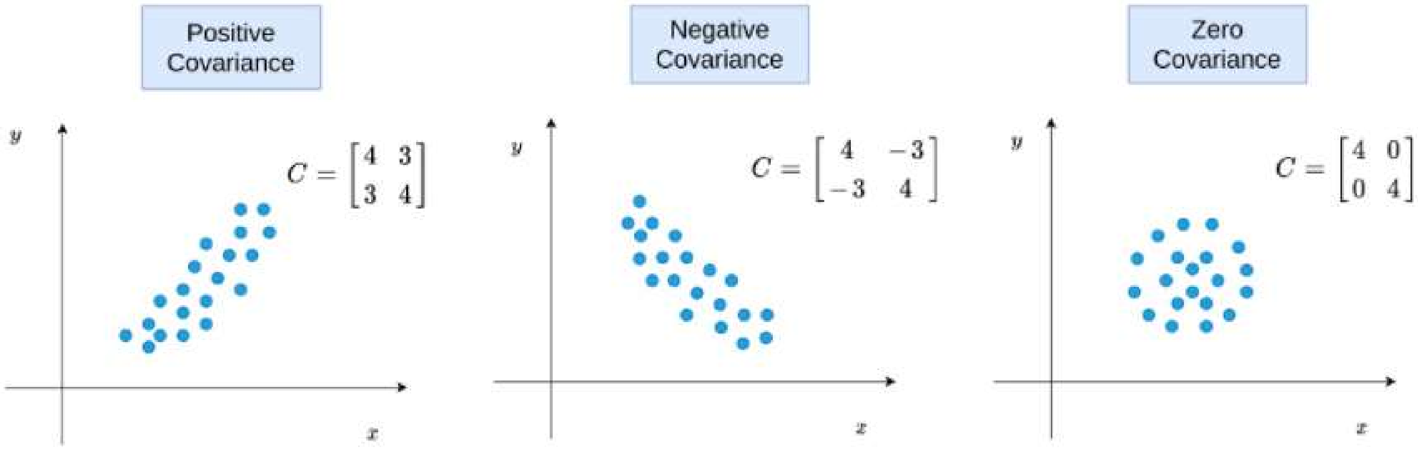
$$\begin{array}{c c} & \begin{array}{cc} x & y \end{array} \\ \begin{array}{c} x \\ y \end{array} & \begin{bmatrix} var(x) & cov(x,y) \\ cov(x,y) & var(y) \end{bmatrix} \end{array} \qquad \begin{array}{c c} & \begin{array}{ccc} x & y & z \end{array} \\ \begin{array}{c} x \\ y \\ z \end{array} & \begin{bmatrix} var(x) & cov(x,y) & cov(x,z) \\ cov(x,y) & var(y) & cov(y,z) \\ cov(x,z) & cov(y,z) & var(z) \end{bmatrix} \end{array}$$

2 and 3- dimensional covariance matrices

6.1. Covariance matrix

Before going further, let's clarify some concepts:

- **Covariance matrix**

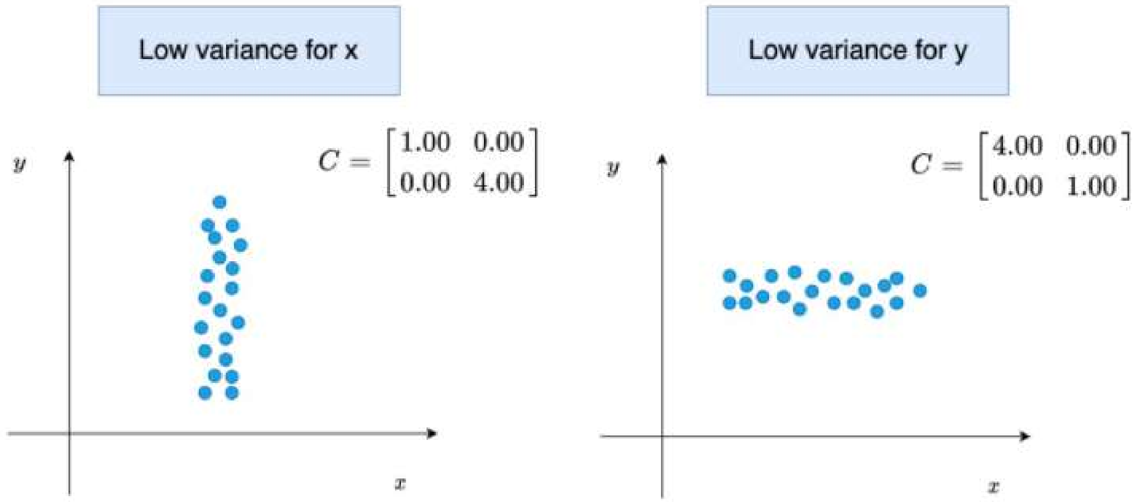


Sergen Cansiz

6.1. Covariance matrix

Before going further, let's clarify some concepts:

- **Covariance matrix**



Sergen Cansiz

Covariance near zero and different variances

6.1.3. Principal Components Analysis (PCA)

- When should I use PCA?

1. Do you want to reduce the number of variables, but **you *are not able to identify variables*** to completely remove from consideration?
2. Do you want ***to ensure your variables are independent*** of one another?
3. Are you ***comfortable making your independent variable less interpretable***?

6.1.3. Principal Components Analysis (PCA)

- PCA introduces a **new set of variables (smaller number of variables)**, called **principal components (PCs)**, by linear combination of the original variables in the data, standardized to zero mean and unit variance.
- **The axes or new variables are the PCs and are ordered by variance:**
  - The first component, *PC 1*, represents the *direction of the highest variance of the data*.
  - The direction of the *PC 2*, represents the *highest of the remaining variance* ***orthogonal*** *to the PC 1*.

This can be naturally **extended to obtain the required number of components, which together span a component space covering the desired amount of variance.**

6.1.3. Covariance matrix – relation with PCA

Before going further, let's clarify some concepts:

- **Eigenvalues and eigenvectors of covariance matrix:**

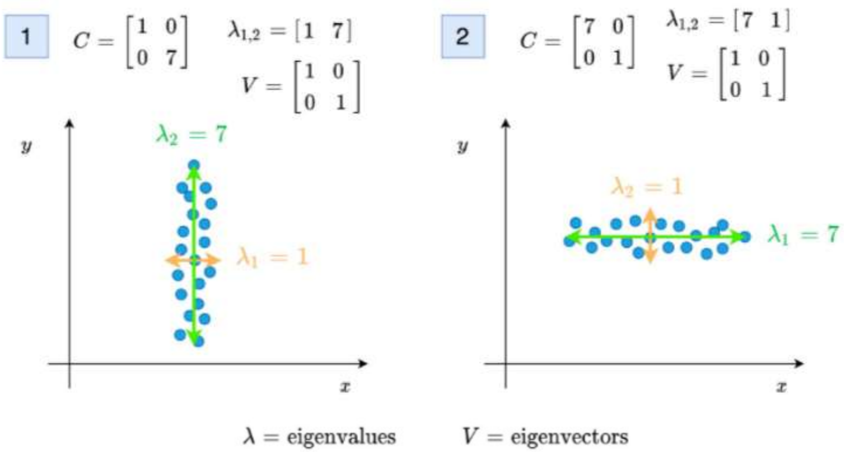
- The **eigenvalues represent the magnitude of the spread** in the direction of the principal components in PCA.

- The **eigenvectors show the direction**.

6.1.3 Covariance matrix – relation with PCA

Before going further, let's clarify some concepts:

- **Eigenvalues and eigenvectors of covariance matrix**

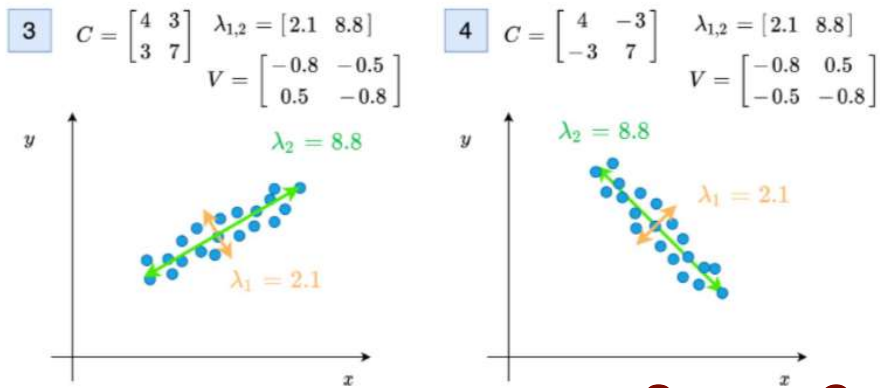


- The first and second plots show the distribution of points when the covariance is near zero (independent variables).

Note: when the covariance is zero the eigenvalues=variance values

- The third and fourth plots represent the distribution of points when the covariance is different from zero.

Note: here we need to calculate the eigenvalues and eigenvectors



6.1.3 Principal Components Analysis (PCA)

PCA is maybe the most popular technique to examine high-dimensional data (unsupervised learning)

PCA computes a rotation matrix : $\boldsymbol{W} \in \mathbb{R}^{\boldsymbol{P}\times\boldsymbol{P}}$ **from the matrix of features** $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{N}\times\boldsymbol{P}}$

W can be understood as a **mapping function that transforms the observations in *X* to a rotated space**

The **coordinates of observations in *X* are transformed to** their new form, **Z**, via: $Z = XW$

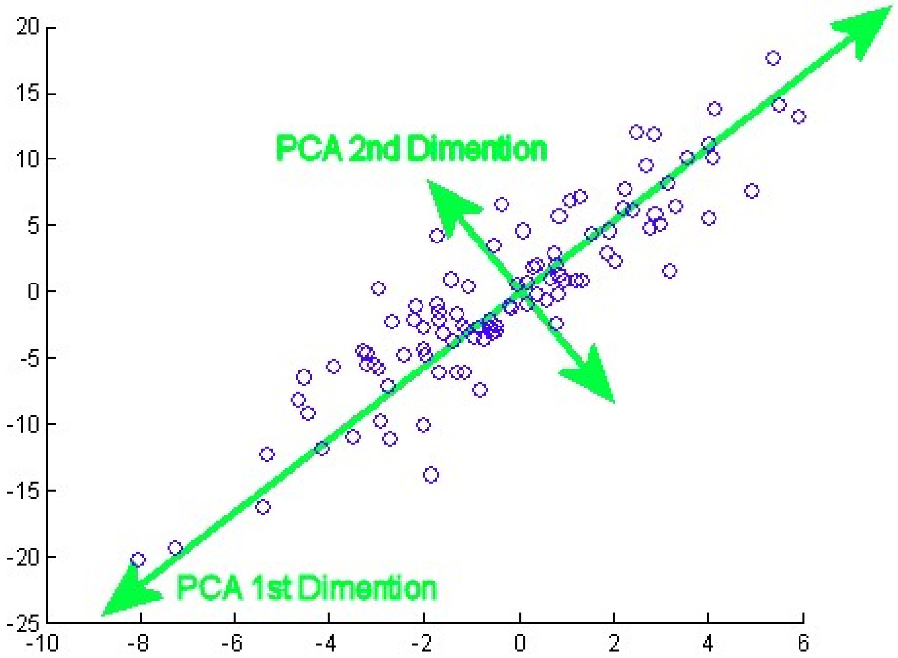
The rotation matrix, *W*, is constructed through orthogonal linear transformations. **Each of these transformations is performed in order to maximize the variance on the data**

6.1.3 Principal Components Analysis (PCA)

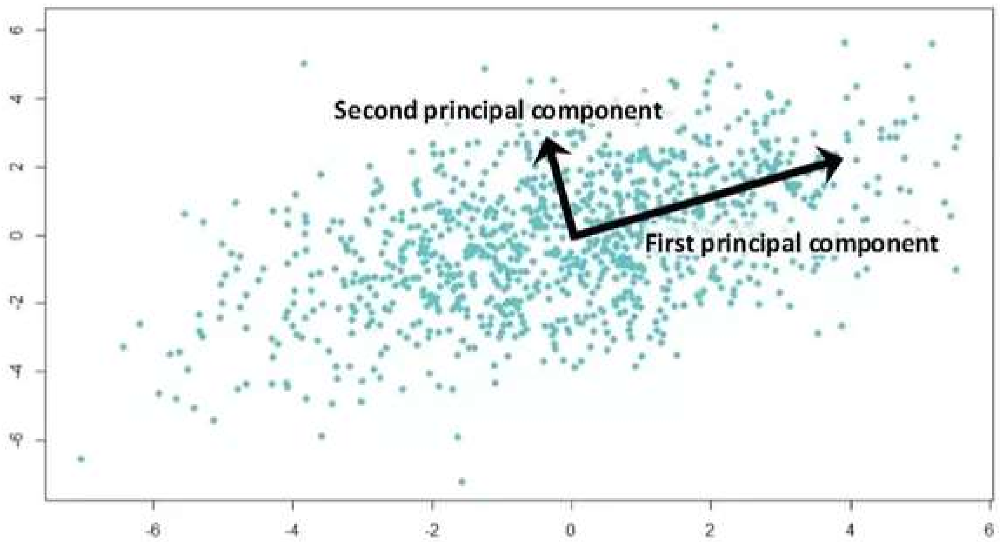
Steps:

1. Take the **matrix of features** $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{N} \times \boldsymbol{P}}$, $N > P$
2. Compute the **mean vector for each dimension**
3. Compute the **covariance matrix**
4. Compute the eigenvectors and corresponding eigenvalues for each dimension
5. Sort the eigenvectors by decreasing eigenvalues and choose ***P*** eigenvectors with the largest eigenvalues to form a new matrix $\boldsymbol{W} \in \mathbb{R}^{\boldsymbol{P} \times \boldsymbol{P}}$
6. Use this eigenvector matrix to transform the samples onto the new subspace: $Z = XW$

6.1.3 Principal Components Analysis (PCA)



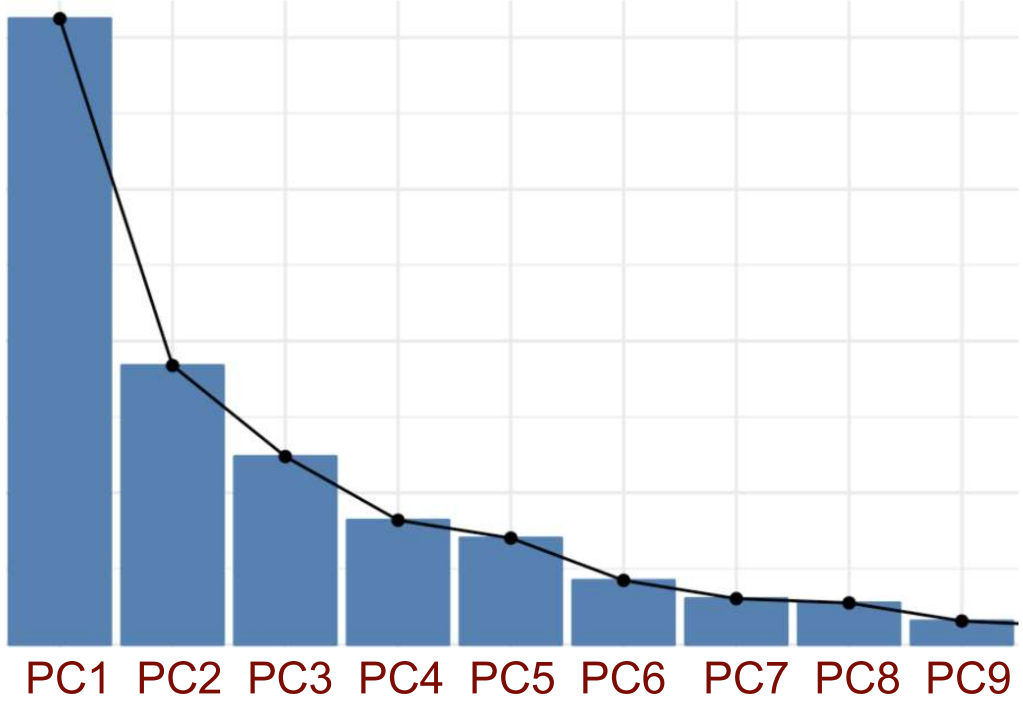
weigend.com



Wavy AI Research Foundation

6.1.3 Principal Components Analysis (PCA)

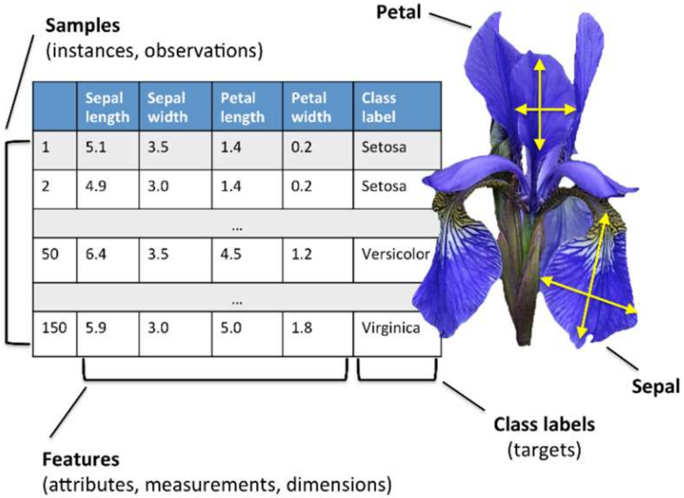
- In highly dimensional datasets, **the vast majority of the variance in the data is often captured by a small number of principal components.**
- A **plot of the distribution of the variance across principal components may look like this:**



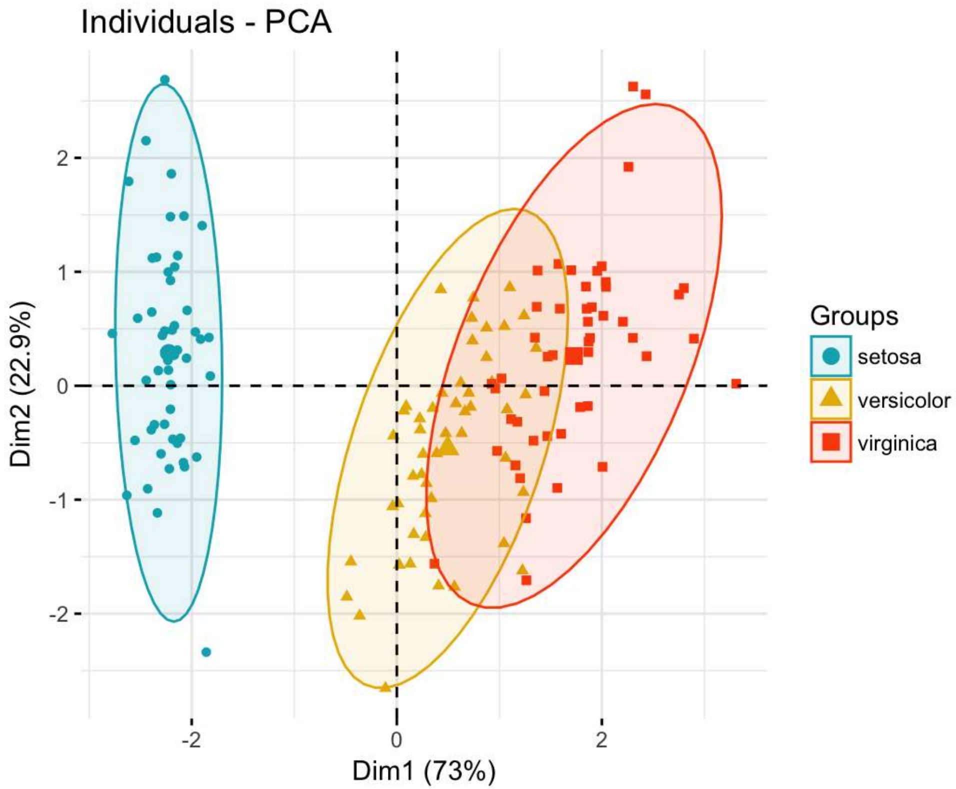
6.1.3 Principal Components Analysis (PCA)


```
##   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
```

3 kind of Iris flowers with 4 attributes: sepal length, sepal width, petal length and petal width



PCA identifies the combination of attributes (PCs, or directions in the feature space) **that account for the most variance in the data.**



Here we plot the different samples on the 2 first PCs.

6.1.3 Principal Components Analysis (PCA)

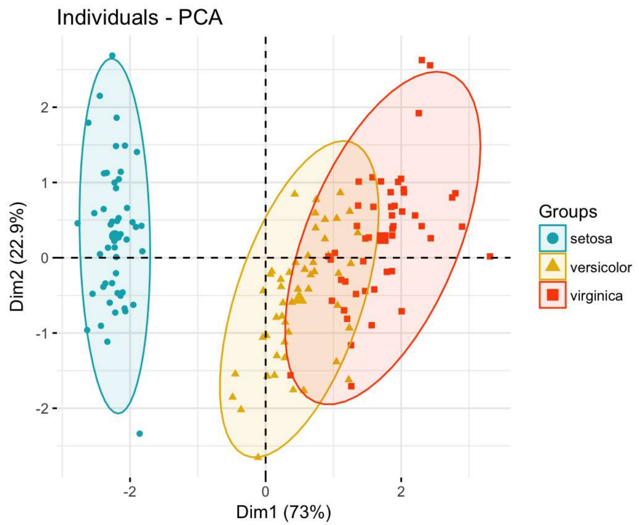
In the theoretical class, we saw:

##	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

3 kind of Iris flowers with 4 attributes: sepal length, sepal width, petal length and petal width

PCA identifies the combination of attributes (PCs, or directions in the feature space) that account for the most variance in the data.

In practice:



How to do PCA Visualization using R:

The following functions, from factoextra package can be used:

`fviz_pca_ind()`: Graph of individuals

`fviz_pca_var()`: Graph of variables

`fviz_pca_biplot()` (or **`fviz_pca()`**): Biplot of individuals and variables

6.1.3. Dimension reduction – PCA

In the theoretical class, we saw:

3 kind of Iris flowers with 4 attributes: sepal length, sepal width, petal length and petal width

```
##   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
```

First, we need to install the packages and load the libraries:

```
> install.packages("devtools")
> library("devtools")
> install.packages(“factoextra")
> library("factoextra")
```

Afterwards, we prepare the dataframe if needed:

The variable Species (index = 5) is removed (not numerical)

We use ‘`center=TRUE`’ to center the variables to 0 and we scale them to have variance 1 by using ‘`scale.=TRUE`’

```
> iris_pca<-prcomp(iris[,-5], center=TRUE, scale.=TRUE)
```

6.1.3. Dimension reduction – PCA

In the theoretical class, we saw:

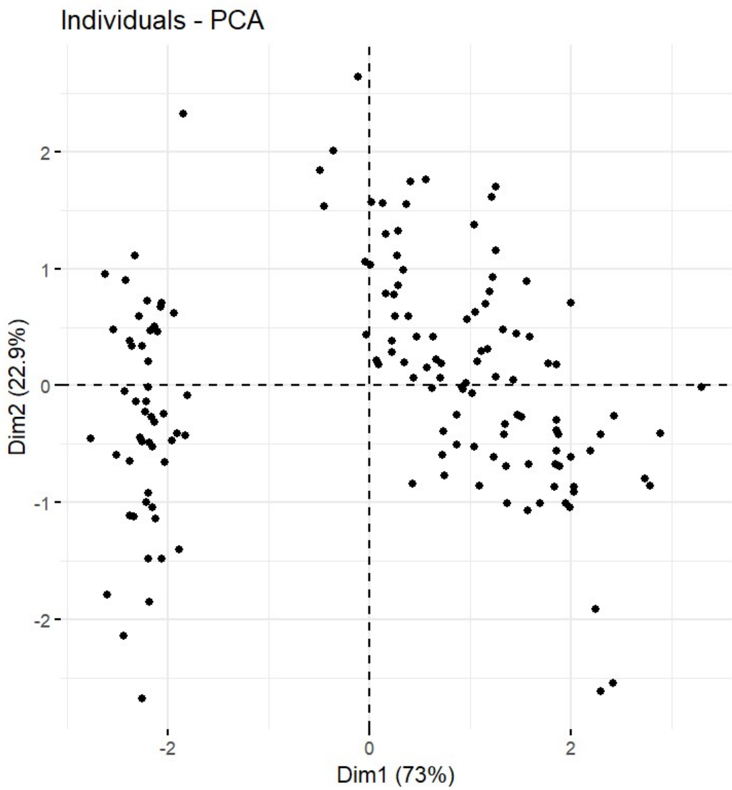
3 kind of Iris flowers with 4 attributes: sepal length, sepal width, petal length and petal width

```
##    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
```

How to do PCA Visualization using R:

fviz_pca_ind(iris_pca, geom="point")

Graph of individuals using only points



6.1.3. Dimension reduction – PCA

In the theoretical class, we saw:

3 kind of Iris flowers with 4 attributes: sepal length, sepal width, petal length and petal width

```
##   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
```

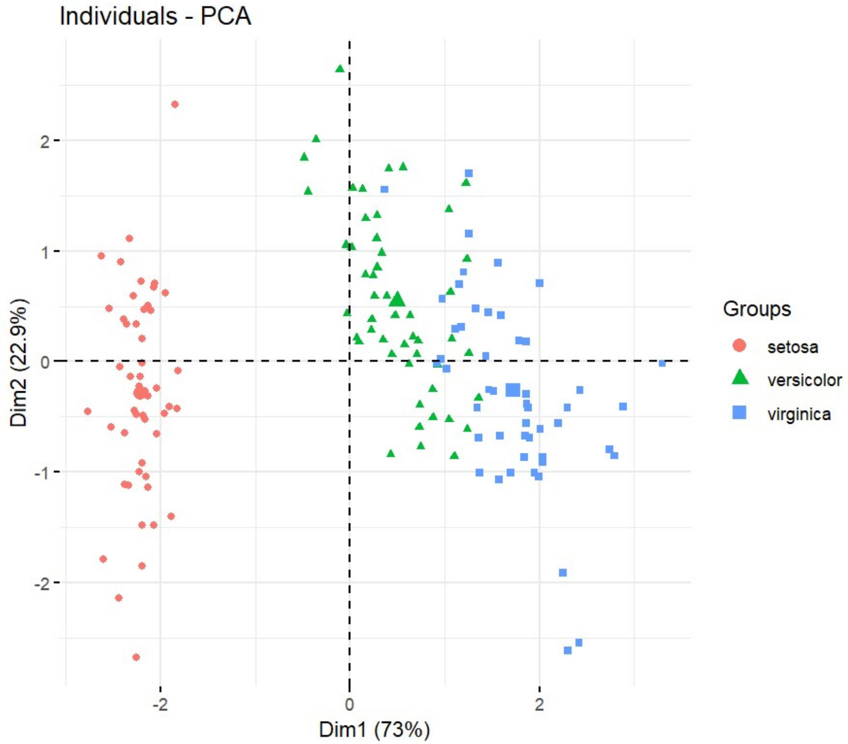
How to do PCA Visualization using R:

fviz_pca_ind(iris_pca, geom="point")

Graph of individuals using only points

fviz_pca_ind(iris_pca, label="none", habillage=iris$Species)

To Color individuals by groups

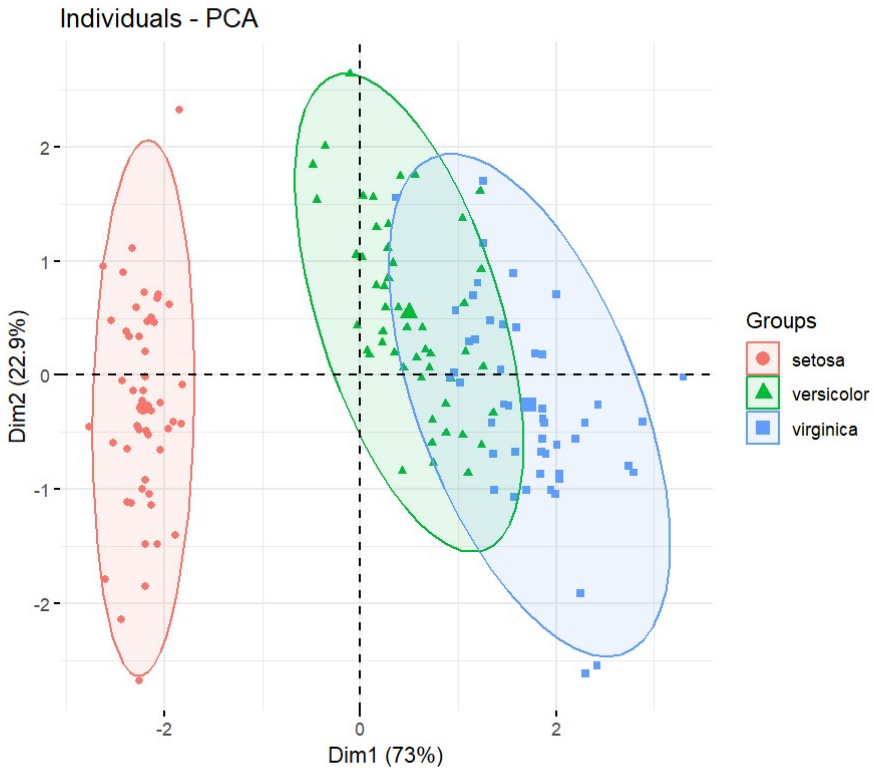


6.1.3. Dimension reduction – PCA

In the theoretical class, we saw:


```
##   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
```

3 kind of Iris flowers with 4 attributes: sepal length, sepal width, petal length and petal width



How to do PCA Visualization using R:

fviz_pca_ind(iris_pca, geom=“point”)

Graph of individuals using only points

fviz_pca_ind(iris_pca, **label="none", habillage=iris$Species)**

To Color individuals by groups

fviz_pca_ind(iris_pca, label="none", habillage=iris$Species, **addEllipses=TRUE, ellipse.level=0.95)**

To add ellipses

6.1.3. Dimension reduction – PCA

In the theoretical class, we saw:

3 kind of Iris flowers with 4 attributes: sepal length, sepal width, petal length and petal width

```
##   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
```

More options in:
http://www.sthda.com/english/wiki/fviz-pca-quick-principal-component-analysis-data-visualization-r-software-and-data-mining

How to do PCA Visualization using R:

`summary(iris_pca)` # Give us the importance of the components

```
> summary(iris_pca)
Importance of components:
                          PC1    PC2     PC3     PC4
Standard deviation     1.7084 0.9560 0.38309 0.14393
Proportion of Variance 0.7296 0.2285 0.03669 0.00518
Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
> 
```

6.1.3 Principal Components Analysis (PCA)

Summarizing:

- PCA is a very interpretable method.
- **Each PC** is well-defined as we know that it **is orthogonal to the other dimensions.**
- **We can obtain the variance that is explained by each PC to select an appropriate number of dimensions**

Weakness of PCA:

It tends to be highly affected by outliers in the data

To overcome this issue many robust versions of PCA has been developed: RandomizedPCA, sparsePCA, etc

PCA works best only with continuous data

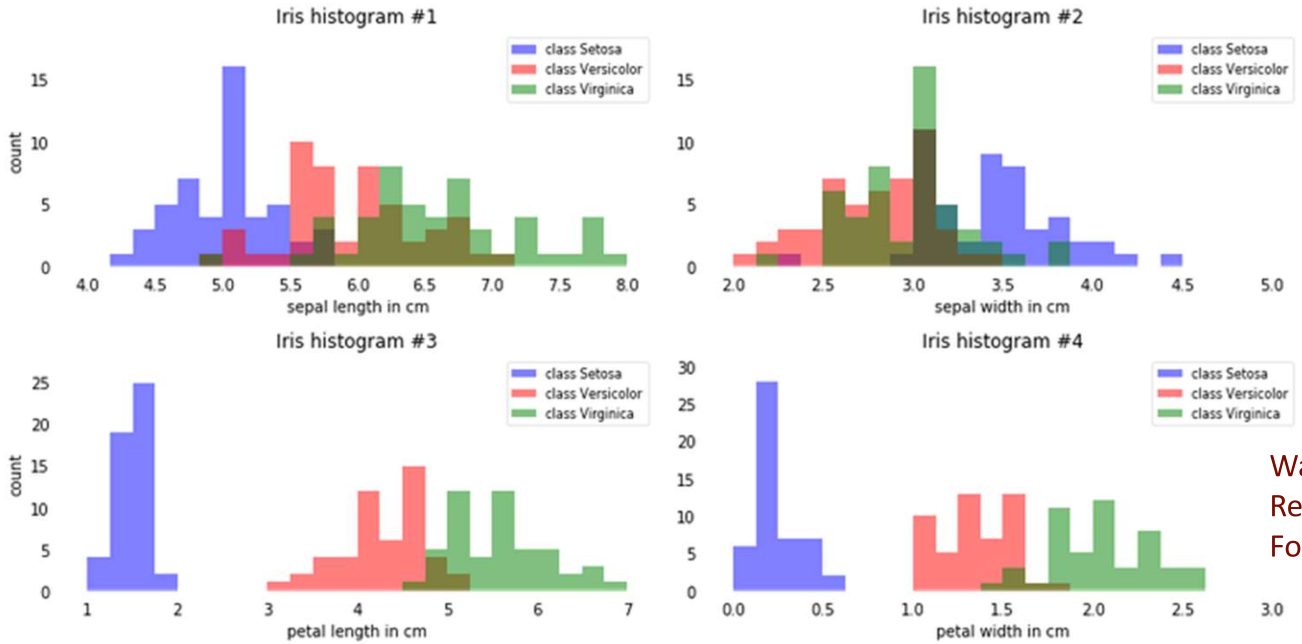
6.1.3 Principal Components Analysis (PCA)

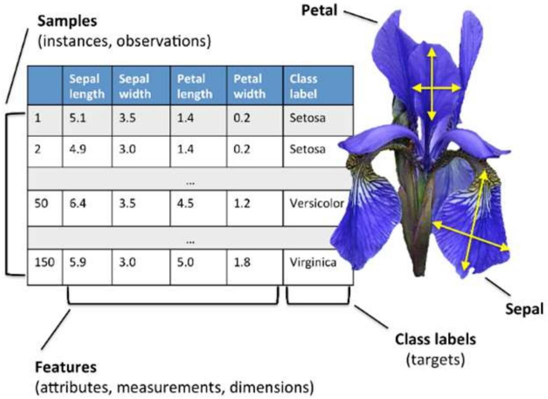

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
```

$$\mathbf{X} = \begin{bmatrix} x1_{\text{sepal length}} & x1_{\text{sepal width}} & x1_{\text{petal length}} & x1_{\text{petal width}} \\ \dots & & & \\ x2_{\text{sepal length}} & x2_{\text{sepal width}} & x2_{\text{petal length}} & x2_{\text{petal width}} \end{bmatrix}, y = \begin{bmatrix} \omega_{\text{iris-setosa}} \\ \dots \\ \omega_{\text{iris-virginica}} \end{bmatrix}$$

En numèriques

3 kind of Iris flowers with 4 attributes: sepal length, sepal width, petal length and petal width





	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

Wavy AI Research Foundation

! Remark: **For low-dimensional datasets like Iris, those histograms would already be very informative.**

6.1.4 Linear Discriminant Analysis (LDA)

LDA seeks to best **separate** (or discriminate) **the samples** in the training dataset **by their class value.**

The fundamental idea of linear combinations goes back as far as the 1960s

✓ **The idea behind LDA: to find a new feature space to project the data in order *to maximize classes separability***

In 1988, the statiscian Ronald Fisher proposed **:**

- Maximize the function that represents the difference between the means, normalized by a measure of the within-class variability

6.1.4 Linear Discriminant Analysis (LDA)

The Fisher's model seeks to **find a linear combination of input variables** that:

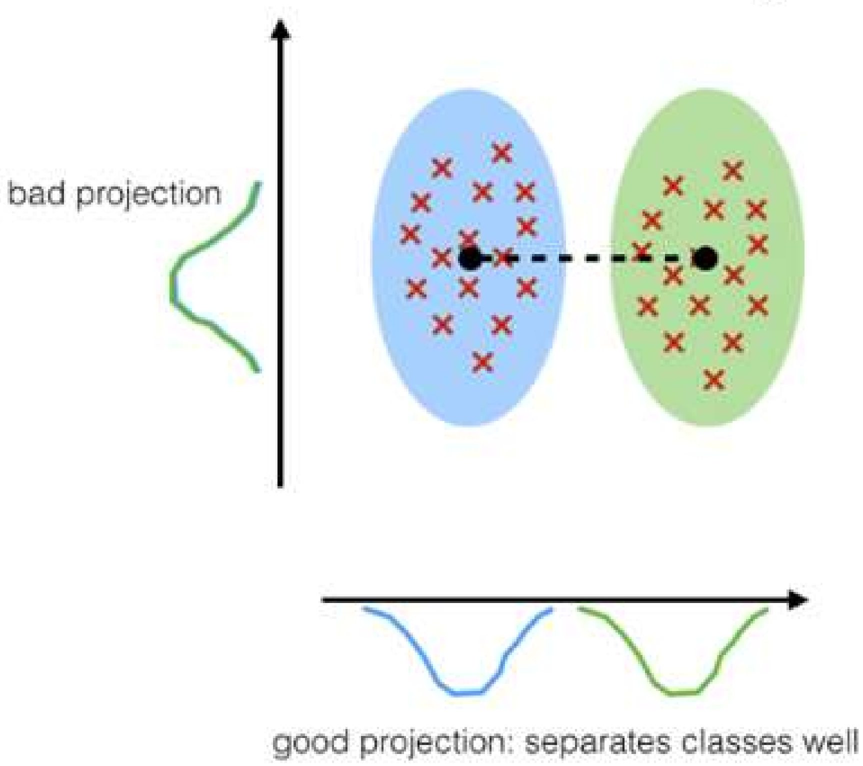
- achieves the **maximum separation** for samples **between classes** (class centroids or means),
- and the **minimum separation** of samples **within each class**.

The LDA takes the mean value for each class and considers variants to make predictions assuming a Gaussian distribution

6.1.4 Linear Discriminant Analysis (LDA)

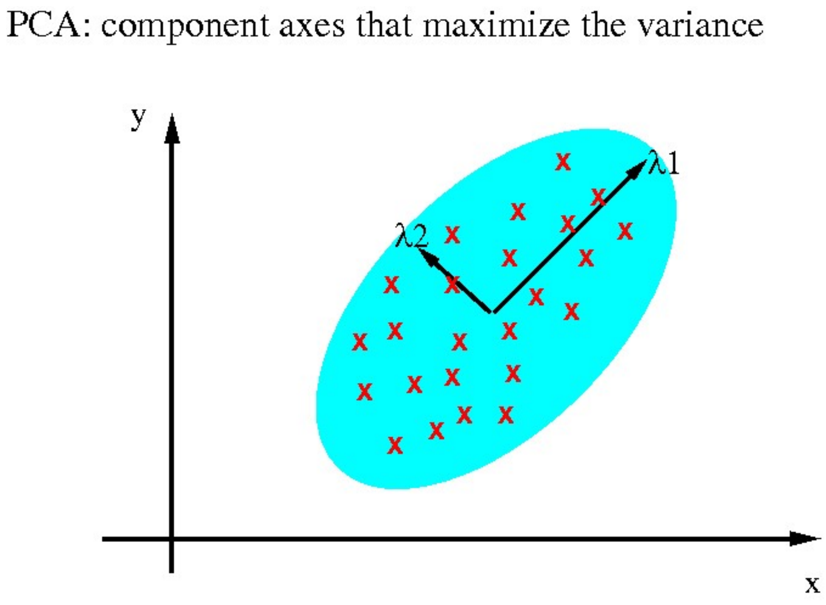
LDA seeks to best **separate** (or discriminate) **the samples** in the training dataset **by their class value.**

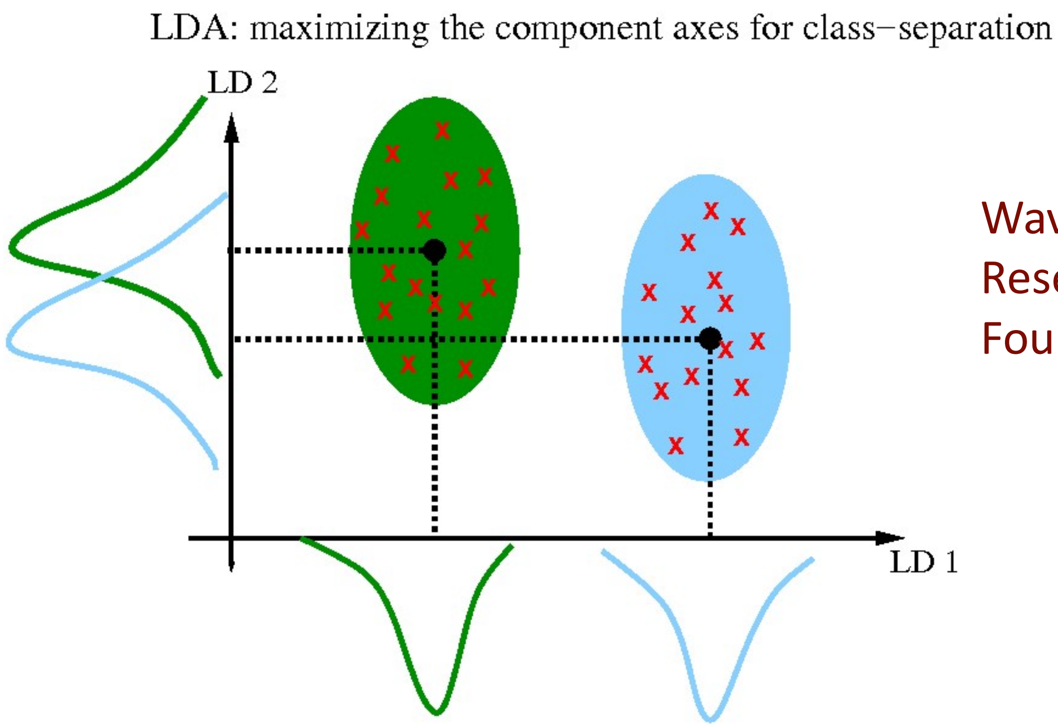
Maximizing the component axes for class-separation:



Wavy AI Research Foundation

6.1.4 PCA versus LDA

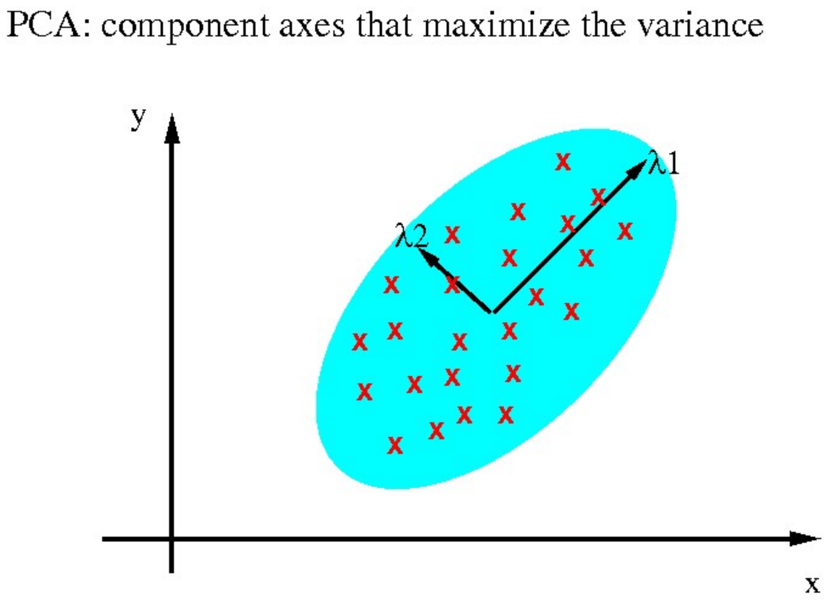


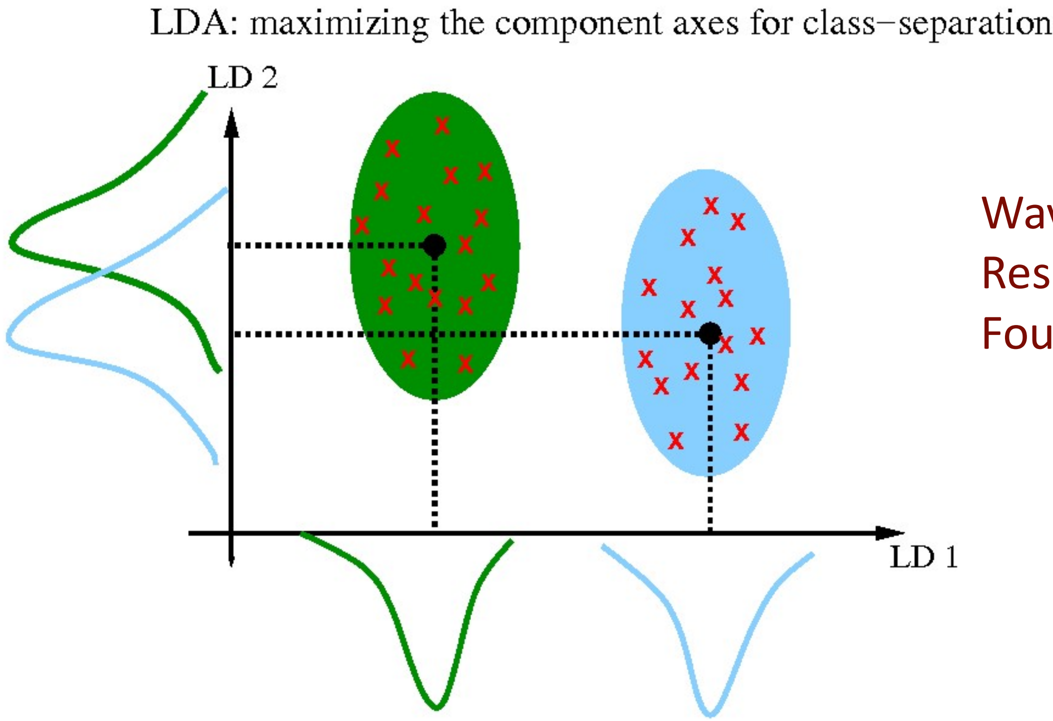


Wavy AI Research Foundation

- Both, LDA and PCA are linear transformation techniques that are commonly used for dimensionality reduction (both are techniques for the data Matrix Factorization)

6.1.4 PCA versus LDA





Wavy AI Research Foundation

- **PCA** is unsupervised algorithm that **attempts to find the orthogonal component axes of maximum variance in a dataset**
- **while the goal of LDA** as supervised algorithm **is to find the feature subspace that optimizes class separability.**

6.1.4. PCA versus LDA

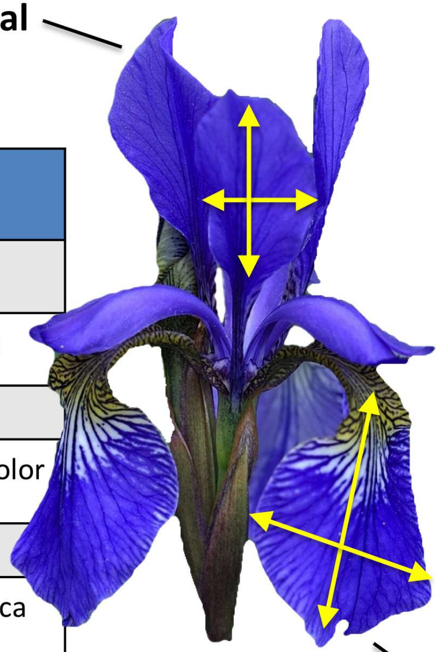
Samples
(instances, observations)

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
…					
50	6.4	3.5	4.5	1.2	Versicolor
…					
150	5.9	3.0	5.0	1.8	Virginica

Features
(attributes, measurements, dimensions)

Class labels
(targets)

Petal



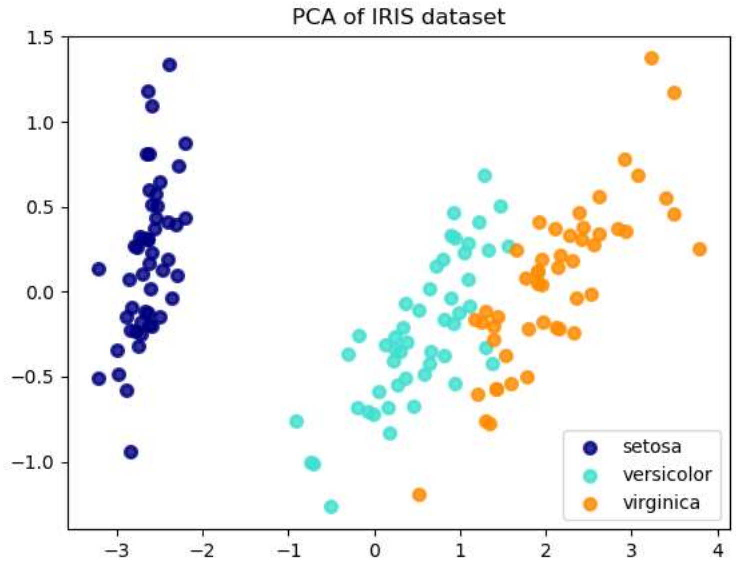
Sepal

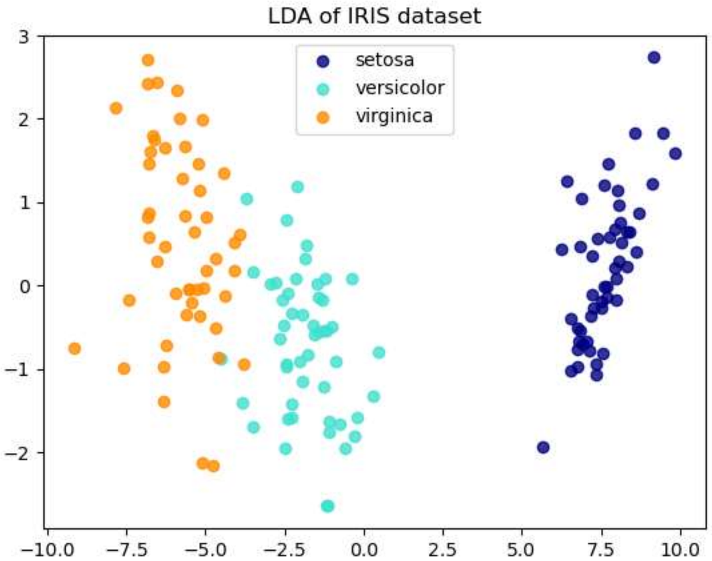

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
```

6.1.4. PCA versus LDA

PCA identifies the combination of attributes (PCs, or directions in the feature space) that account for the **most variance in the data.**

3 kind of Iris flowers with 4 attributes: sepal length, sepal width, petal length and petal width





LDA: tries to **identify attributes** that account for the most **variance between classes**

6.1.4. Linear Discriminant Analysis (LDA)

Steps: (see this link for an example with iris dataframe)

1. Compute the **d-dimensional mean vector for the different classes** from the dataset. **(in PCA was for each direction)**
2. Compute the Scatter matrix (in between class and within the class scatter matrix)
3. Sort the Eigen Vector by decreasing Eigen Value order and choose *k* eigenvector with the largest eigenvalue to from a *dxk* dimensional matrix W (where every column represent an eigenvector)
4. Used *dxk* eigenvector matrix to transform the sample onto the new subspace. This can be summarised by the matrix multiplication:

$$Y = XW$$

where *X* is a *nxd* dimension matrix representing the *n* samples and you are transformed *nxk* dimensional samples in the new subspace.

6.1.4. Linear Discriminant Analysis (LDA)

LDA **can be useful in areas like image recognition and predictive analysis in marketing**

Weakness of LDA:

- LDA **does not work well if the design is not balanced** (i.e. the number of objects in various classes are (highly) different)
- If the **distribution of your data is significantly non-Gaussian**, the LDA **might not perform very well**.
- It is **sensitive to overfit**
- **LDA is not applicable (inferior) for non-linear problems**

Thanks for your attention!

Judit Chamorro Servent

Departament de Matemàtiques

judit.chamorro@uab.cat