

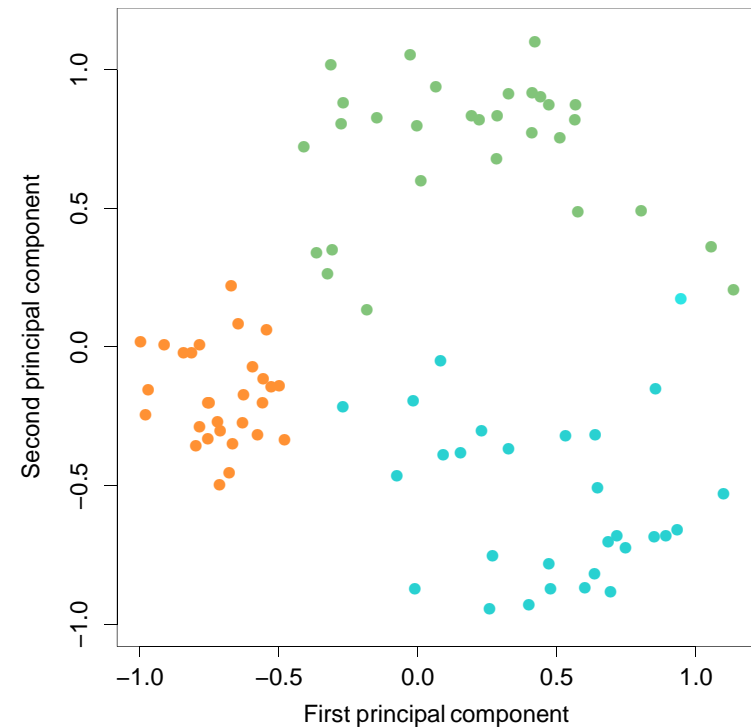
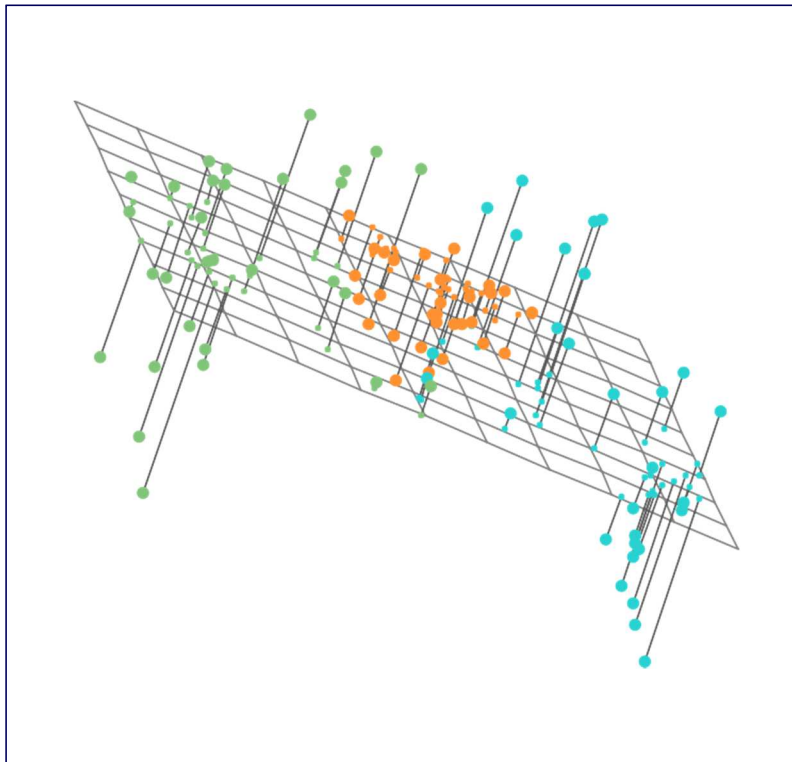
# Phân tích thành phần chính

## Principal Component Analysis (PCA)

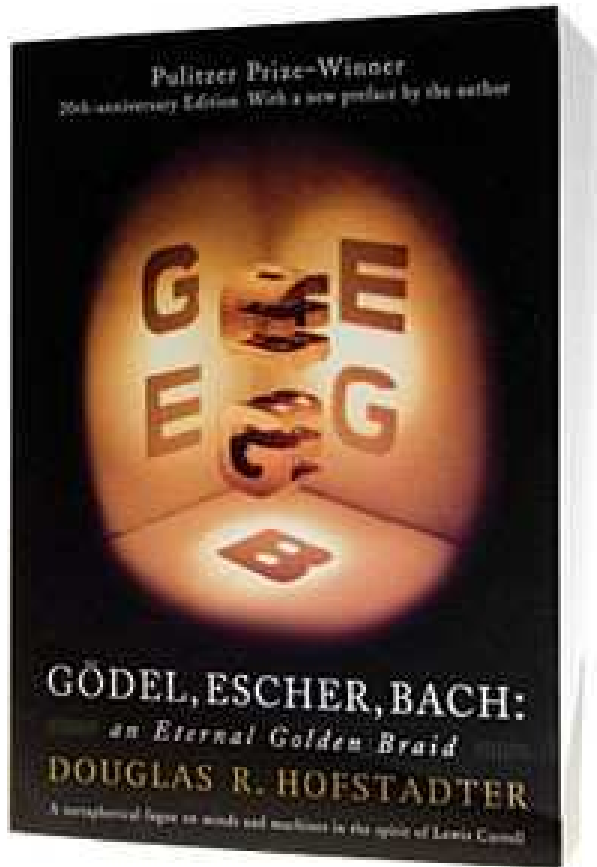
*Các slides ở mục này tham khảo từ bài giảng của GS. Nguyễn Văn Tuấn*



# Giảm chiều dữ liệu



# Phép chiếu



# Phân tích thành phần chính

- Đề xướng bởi Pearson (1901) và Hotelling (1933)
- Mô tả sự biến thiên của nhiều biến số có liên quan nhau bằng ***một nhóm biến số mới không liên quan đến nhau***

# Phân tích thành phần chính

- Nếu hai biến (hay 2 items) có tương quan với nhau
  - Chúng có thể phản ánh một hiện tượng tiềm ẩn (hay một yếu tố không quan sát được – latent factor)
  - Nếu chúng phản ánh một latent variable, thì tổng hợp chúng thành 1 biến là hợp lí
- Các biến latent variables còn gọi là "**factors**" hay "**principal components**"

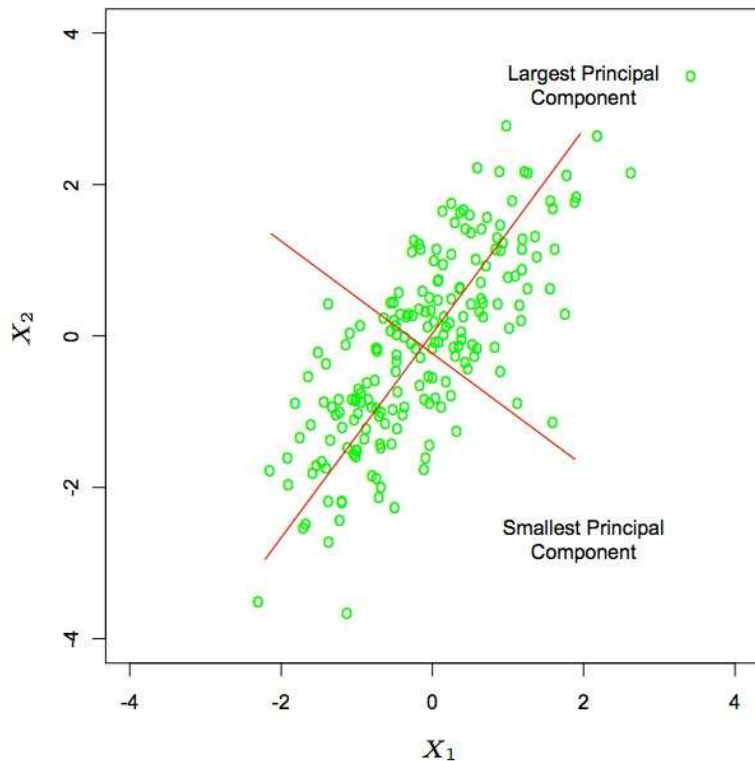
# Phân tích thành phần chính

- Trong không gian mới, các liên kết tiềm ẩn của dữ liệu có thể được khám phá
- Ví dụ: Thị trường ta quan tâm có hàng ngàn mã cổ phiếu làm cách nào để khi quan sát dữ liệu từ hàng ngàn cổ phiếu này ta hình dung được xu hướng của toàn thị trường...

*nguồn: <http://phvu.net/>*



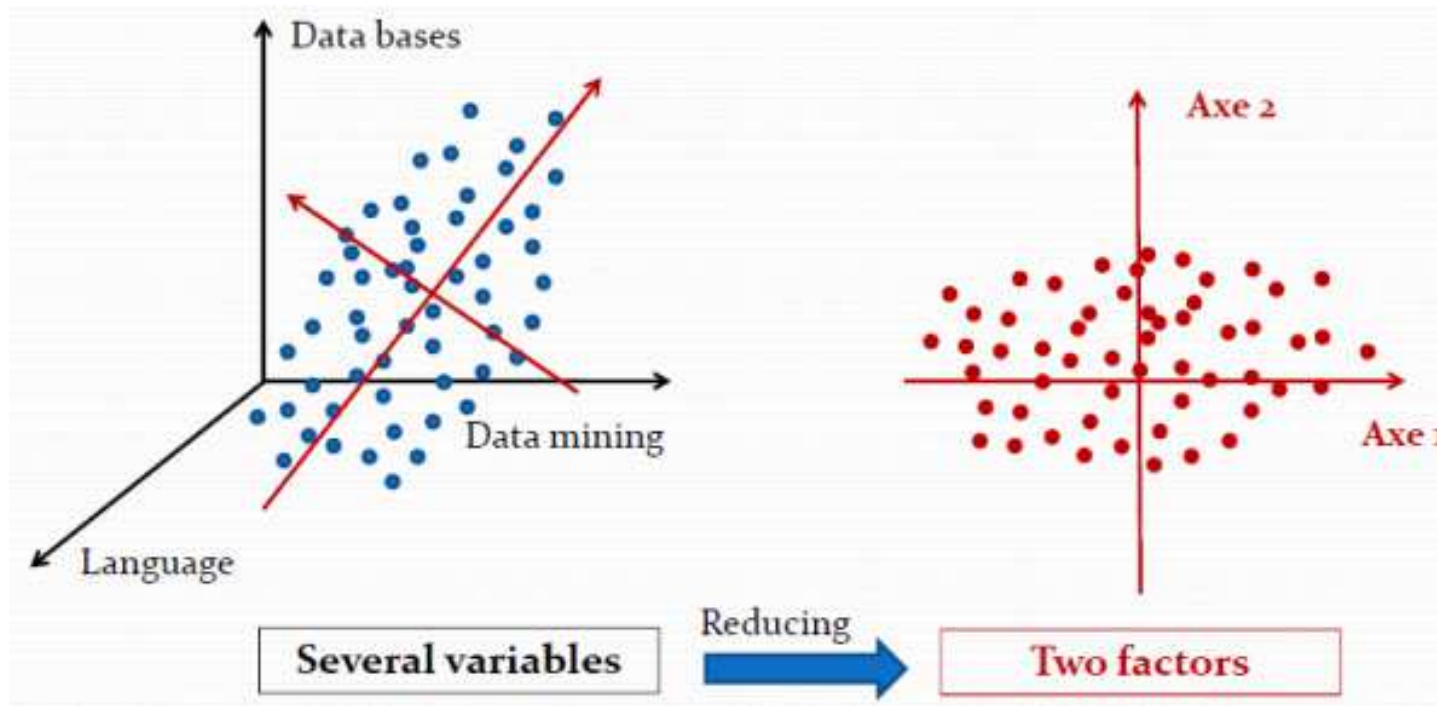
# Phân tích thành phần chính



Minh họa PCA: phép chiếu lên các trục tọa độ khác nhau có thể cho cách nhìn rất khác nhau về cùng một dữ liệu.

nguồn: <http://phvu.net/>

# Phân tích thành phần chính



Giả sử tập dữ liệu ban đầu (tập điểm màu xanh) được quan sát trong không gian 3 chiều (trục màu đen) như hình bên trái. Rõ ràng 3 trục này không biểu diễn được tốt nhất mức độ biến thiên của dữ liệu. PCA do đó sẽ tìm hệ trục tọa độ mới (là hệ trục màu đỏ trong hình bên trái). Sau khi tìm được không gian mới, dữ liệu sẽ được chuyển sang không gian này để được biểu diễn như trong hình bên phải. Rõ ràng hình bên phải chỉ cần 2 trục tọa độ nhưng biểu diễn tốt hơn độ biến thiên của dữ liệu so với hệ trục 3 chiều ban đầu.

nguồn: <http://phvu.net/>



S.Length	S.Width	P.Length	P.Width	Species
5.1		3.5	1.4	0.2 I.setosa
4.9		3	1.4	0.2 I.setosa
4.7		3.2	1.3	0.2 I.setosa
4.6		3.1	1.5	0.2 I.setosa
5		3.6	1.4	0.2 I.setosa
5.4		3.9	1.7	0.4 I.setosa
4.6		3.4	1.4	0.3 I.setosa
5		3.4	1.5	0.2 I.setosa
4.4		2.9	1.4	0.2 I.setosa
5.1		3.8	1.9	0.4 I.setosa
4.8		3	1.4	0.3 I.setosa
5.1		3.8	1.6	0.2 I.setosa
4.6		3.2	1.4	0.2 I.setosa
5.3		3.7	1.5	0.2 I.setosa
5		3.3	1.4	0.2 I.setosa
7		3.2	4.7	1.4 I.versicolor
6.4		3.2	4.5	1.5 I.versicolor
6.9		3.1	4.9	1.5 I.versicolor
5.5		2.3	4	1.3 I.versicolor
6.5		2.8	4.6	1.5 I.versicolor
5.7		2.8	4.5	1.3 I.versicolor
6.3		3.3	4.7	1.6 I.versicolor
4.9		2.4	3.3	1 I.versicolor
6.6		2.9	4.6	1.3 I.versicolor
5.2		2.7	3.9	1.4 I.versicolor
5		2	3.5	1 I.versicolor
6.2		2.8	4.8	1.8 I.virginica
6.1		3	4.9	1.8 I.virginica
6.4		2.8	5.6	2.1 I.virginica
7.2		3	5.8	1.6 I.virginica
7.4		2.8	6.1	1.9 I.virginica
7.9		3.8	6.4	2 I.virginica

# Iris data

# Bối cảnh và dữ liệu

- Bối cảnh: chúng ta có một ma trận dữ liệu gồm  $n$  hàng và  $p$  biến số  $x_1, x_2, \dots, x_p$
- PCA tìm cách hoán chuyển các  $x_i$  thành  $p$  biến mới ( $y_i$ ) nhưng không có liên quan với nhau!

# Khi các biến liên quan đến nhau

- Cách đơn giản nhất là chỉ lưu lại 1 biến duy nhất (bỏ các biến còn lại) – không hợp lí
- Cho trọng số mỗi biến. Trọng số nào?
- Tiêu chuẩn nào?

# Khi các biến liên quan đến nhau

- Tìm phương pháp hoán chuyển ma trận  $\mathbf{X}$  ( $n \times p$ ) sao cho

$$Y = \boldsymbol{\delta}^T \mathbf{X} = \delta_1 X_1 + \delta_2 X_2 + \dots + \delta_p X_p$$

Trong đó:  $\boldsymbol{\delta} = (\delta_1, \delta_2, \dots, \delta_p)^T$  là cột vector gồm trọng số sao cho:

$$\delta_1^2 + \delta_2^2 + \dots + \delta_p^2 = 1$$

# Tiêu chuẩn

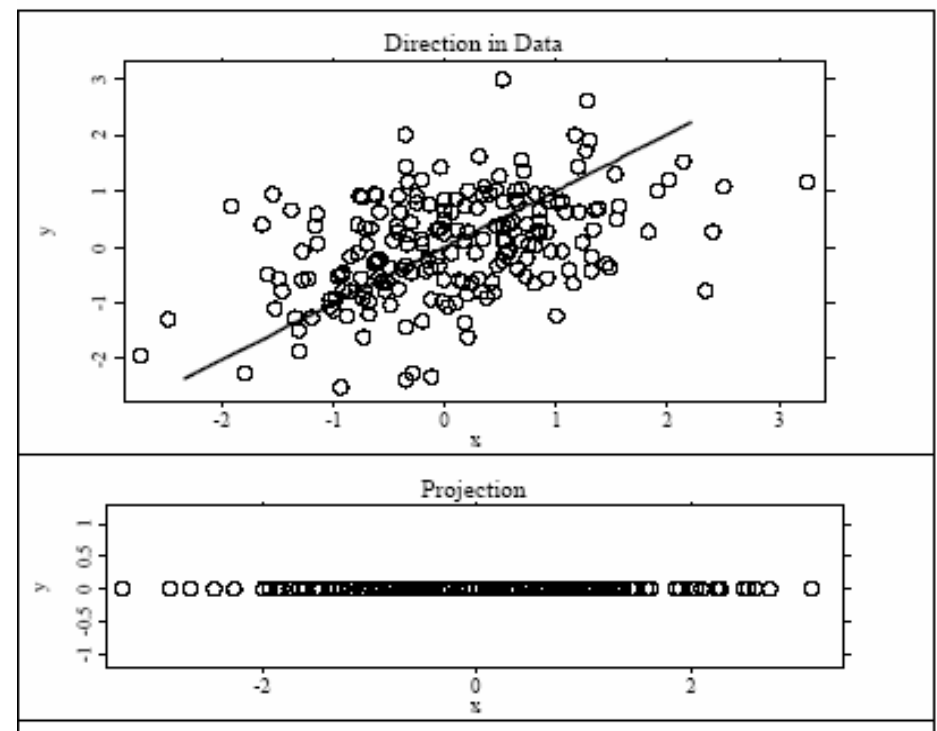
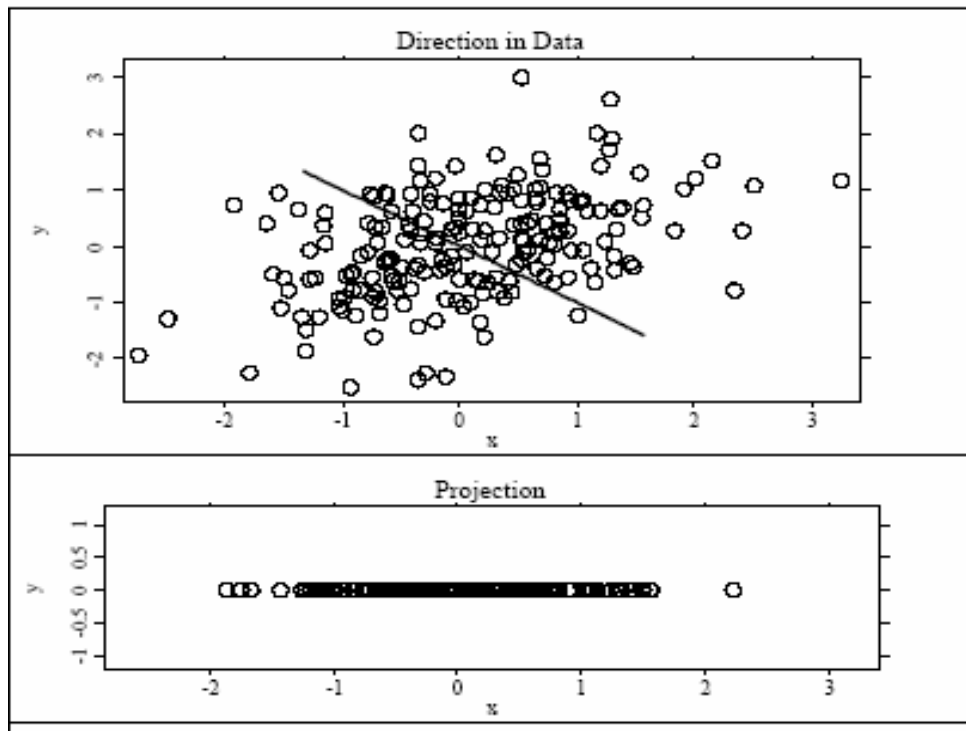
- Tối đa hoá phương sai của dữ liệu dựa trên các biến  $Y$
- Tìm  $\delta$  sao cho

$$\text{Var}(\delta^T \mathbf{X}) = \delta^T \text{Var}(\mathbf{X}) \delta \text{ tối đa}$$

- Ma trận  $\mathbf{C} = \text{Var}(\mathbf{X})$  là hiệp biến (covariance) của các biến  $X_i$



# Thử tưởng tượng ...



# Variance-covariance matrix

$$\begin{pmatrix} v(x_1) & c(x_1, x_2) & \dots\dots\dots c(x_1, x_p) \\ c(x_1, x_2) & v(x_2) & \dots\dots\dots c(x_2, x_p) \\ \vdots & \vdots & \ddots & \vdots \\ c(x_1, x_p) & c(x_2, x_p) & \dots\dots\dots v(x_p) \end{pmatrix}$$

# Có nghĩa là chúng ta tìm ...

- Hướng của  $\delta$  được xác định bởi véc tơ riêng (vector eigen)  $\gamma_1$  tương đương với giá trị riêng (eigen) lớn nhất của ma trận  $\mathbf{C}$
- Vector thứ 2 cũng trực giao (orthogonal, tức không liên quan) với vector 1
- v.v.



# Do đó, PCA cung cấp

- Một nhóm biến mới ( $Y_i$ ) là hàm số tuyến tính của các biến  $X_i$ :

$$Y_i = a_{i1}x_1 + a_{i2}x_2 + \dots + a_{ip}x_p; i = 1, 2, \dots, p$$

- Biến mới  $Y_i$  được tạo ra theo mức độ quan trọng nhưng suy giảm theo độ quan trọng
- Chúng được gọi là "**principal components**"

# Tính toán eigenvalue và eigenvector

- Giá trị eigen  $\lambda_i$  được xác định bằng cách giải phương trình

$$\det(C - \lambda I) = 0$$

- Vector eigen là những cột của ma trận A với đặc điểm

$$C = A D A^T$$

- Trong đó

$$D = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ 0 & & & \\ 0 & \dots & \dots & \lambda_p \end{pmatrix}$$

# Diễn giải PCA

- Những biến mới (Principal Components) có phương sai bằng giá trị eigen:

$$\text{Var}(Y_i) = \lambda_i \text{ cho tất cả } i = 1, 2, \dots, p$$

- Giá trị  $\lambda_i$  nhỏ  $\Leftrightarrow$  phương sai thấp  $\Leftrightarrow$  dữ liệu thay đổi nhỏ về hướng của  $Y_i$
- Mức độ quan trọng của mỗi PC given by  $\lambda_i / \sum \lambda_i$

# Cần bao nhiêu PC?

- Số PCs sao cho tỉ lệ phương sai giải thích được từ 50-70%
- **Kaiser criterion**: giữ PCs với eigenvalues  $>1$
- **Scree plot**: thể hiện khả năng PC giải thích phương sai của dữ liệu

# Diễn giải ý nghĩa của PC

- Trọng số của biến số trong mỗi PC:
- Nếu  $Y_1 = 0.89X_1 + 0.15X_2 - 0.77X_3 + 0.51X_4$
- Thì  $X_1$  và  $X_3$  có trọng số cao nhất, và là biến quan trọng nhất
- Xem mối tương quan giữa các biến  $X_i$  và PC

# Các bước phân tích

1. Chuẩn bị dữ liệu (chuẩn hoá dữ liệu)
2. Tính ma trận covariance hoặc correlation
3. Tính eigenvalues của ma trận covariance
4. Chọn components



# Dữ liệu của Fisher

S.Length	S.Width	P.Length	P.Width	Species
5.1	3.5	1.4	0.2	I.setosa
4.9	3	1.4	0.2	I.setosa
4.7	3.2	1.3	0.2	I.setosa
4.6	3.1	1.5	0.2	I.setosa
5	3.6	1.4	0.2	I.setosa
5.4	3.9	1.7	0.4	I.setosa
4.6	3.4	1.4	0.3	I.setosa
5	3.4	1.5	0.2	I.setosa
4.4	2.9	1.4	0.2	I.setosa
5.1	3.8	1.9	0.4	I.setosa
4.8	3	1.4	0.3	I.setosa
5.1	3.8	1.6	0.2	I.setosa
4.6	3.2	1.4	0.2	I.setosa
5.3	3.7	1.5	0.2	I.setosa
5	3.3	1.4	0.2	I.setosa
7	3.2	4.7	1.4	I.versicolor
6.4	3.2	4.5	1.5	I.versicolor
6.9	3.1	4.9	1.5	I.versicolor
5.5	2.3	4	1.3	I.versicolor
6.5	2.8	4.6	1.5	I.versicolor
5.7	2.8	4.5	1.3	I.versicolor
6.3	3.3	4.7	1.6	I.versicolor
4.9	2.4	3.3	1	I.versicolor
6.6	2.9	4.6	1.3	I.versicolor
5.2	2.7	3.9	1.4	I.versicolor
5	2	3.5	1	I.versicolor
6.2	2.8	4.8	1.8	I.virginica
6.1	3	4.9	1.8	I.virginica
6.4	2.8	5.6	2.1	I.virginica
7.2	3	5.8	1.6	I.virginica
7.4	2.8	6.1	1.9	I.virginica
7.9	3.8	6.4	2	I.virginica

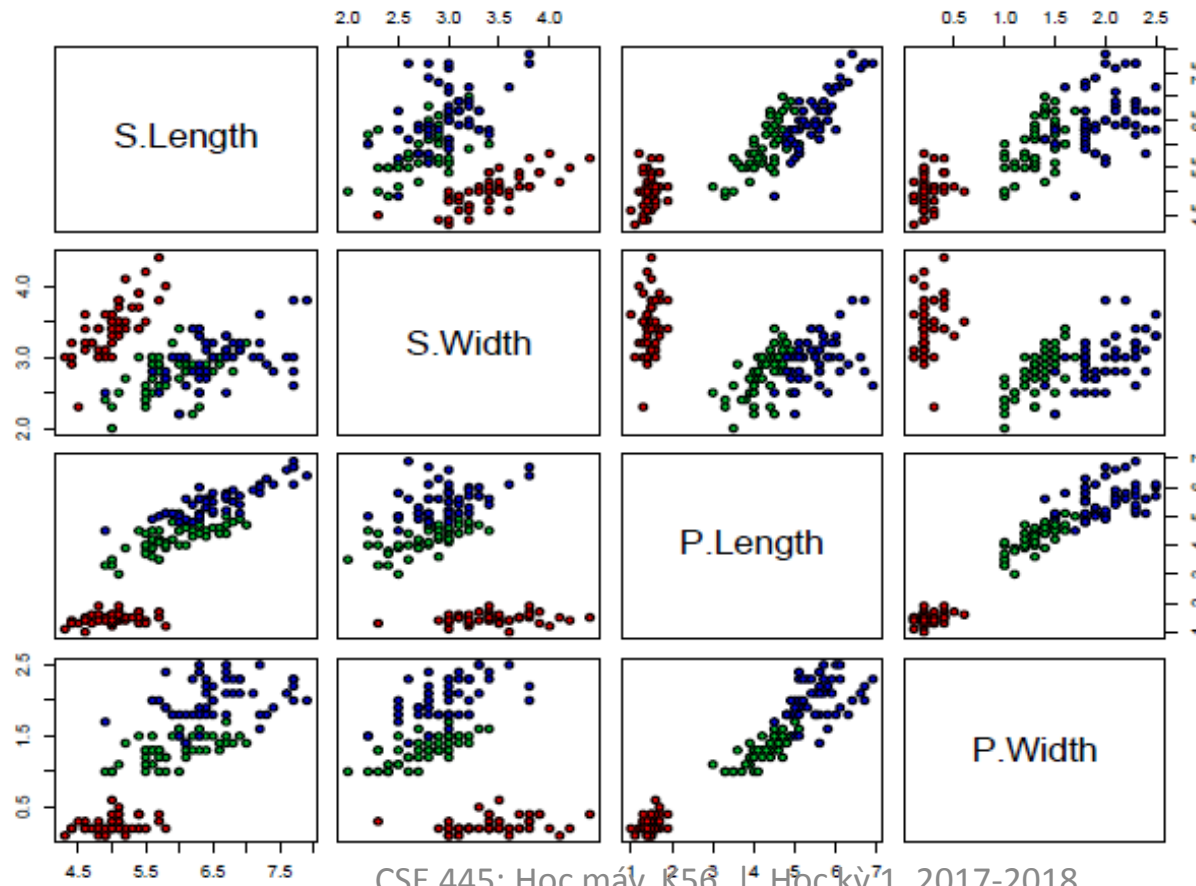
```
data(iris)
head(iris)
```

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



# Quan sát dữ liệu

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



CSE 445: Học máy, K56 | Học kỳ 1, 2017-2018





# Bước 1: Chuẩn hoá data

```
> s.iris = scale(iris[1:4], center = TRUE, scale = TRUE)
```

```
> head(s.iris)
```

	S.Length	S.Width	P.Length	P.Width
[1,]	-0.8976739	1.01560199	-1.335752	-1.311052
[2,]	-1.1392005	-0.13153881	-1.335752	-1.311052
[3,]	-1.3807271	0.32731751	-1.392399	-1.311052
[4,]	-1.5014904	0.09788935	-1.279104	-1.311052
[5,]	-1.0184372	1.24503015	-1.335752	-1.311052
[6,]	-0.5353840	1.93331463	-1.165809	-1.048667

## Bước 2: Tính toán hệ số tương quan

```
> s.corr = cor(s.iris)
> round(s.corr, 2)
```

	S.Length	S.Width	P.Length	P.Width
S.Length	1.00	-0.12	0.87	0.82
S.Width	-0.12	1.00	-0.43	-0.37
P.Length	0.87	-0.43	1.00	0.96
P.Width	0.82	-0.37	0.96	1.00

# Bước 3: Tính toán eigenvalues

```
> eigen = eigen(s.corr)
```

```
> eigen
```

```
$values
```

```
[1] 2.91849782 0.91403047 0.14675688 0.02071484
```

```
$vectors
```

	[ ,1]	[ ,2]	[ ,3]	[ ,4]
[1,]	0.5210659	-0.37741762	0.7195664	0.2612863
[2,]	-0.2693474	-0.92329566	-0.2443818	-0.1235096
[3,]	0.5804131	-0.02449161	-0.1421264	-0.8014492
[4,]	0.5648565	-0.06694199	-0.6342727	0.5235971



## Bước 4: Phân tích với **prcomp**

```
> pca = prcomp(iris[1:4], center=T, scale=T)
> pca
```

Standard deviations:

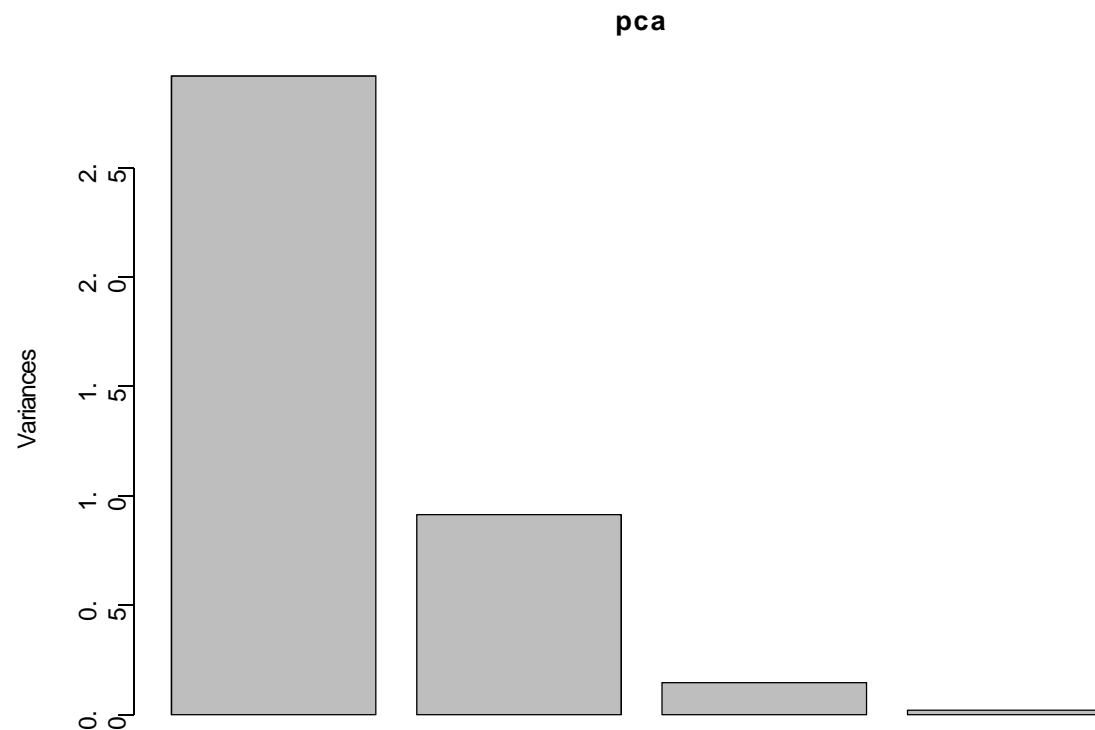
```
[1] 1.7083611 0.9560494 0.3830886 0.1439265
```

Rotation:

	PC1	PC2	PC3	PC4
S.Length	0.5210659	-0.37741762	0.7195664	0.2612863
S.Width	-0.2693474	-0.92329566	-0.2443818	-0.1235096
P.Length	0.5804131	-0.02449161	-0.1421264	-0.8014492
P.Width	0.5648565	-0.06694199	-0.6342727	0.5235971

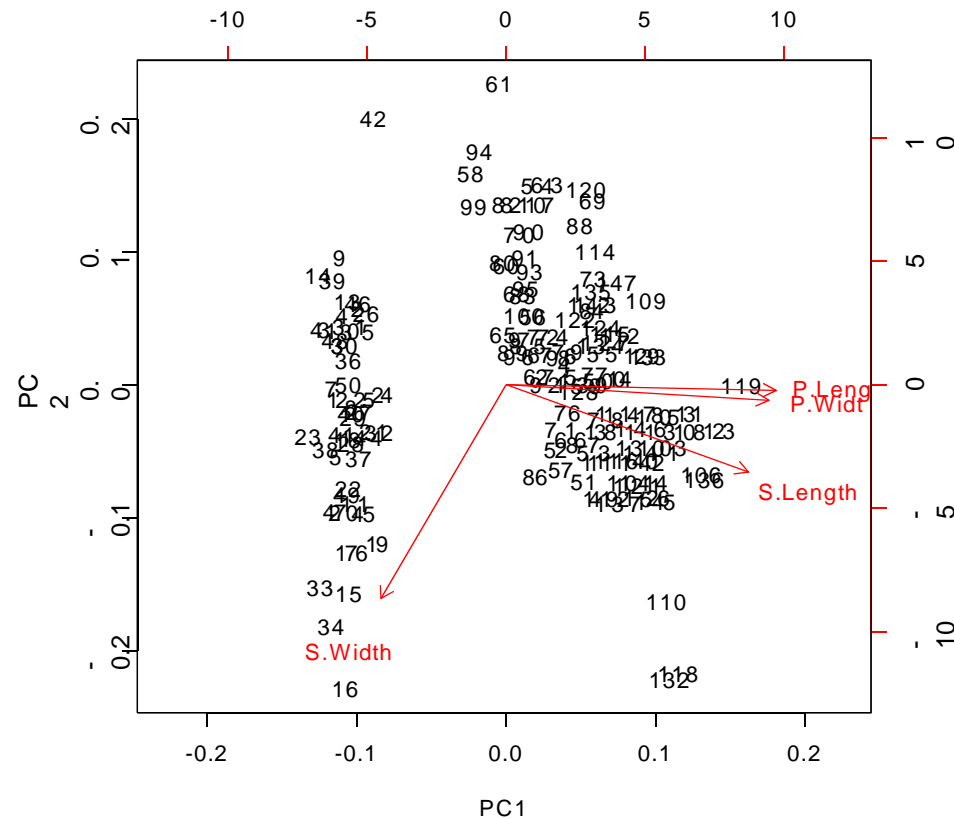
# Bước 4: Phân tích với **prcomp**

```
plot(pca)
```



# Bước 4: Phân tích với **prcomp**

```
biplot(pca)
```



# Bước 4: Phân tích với **prcomp**

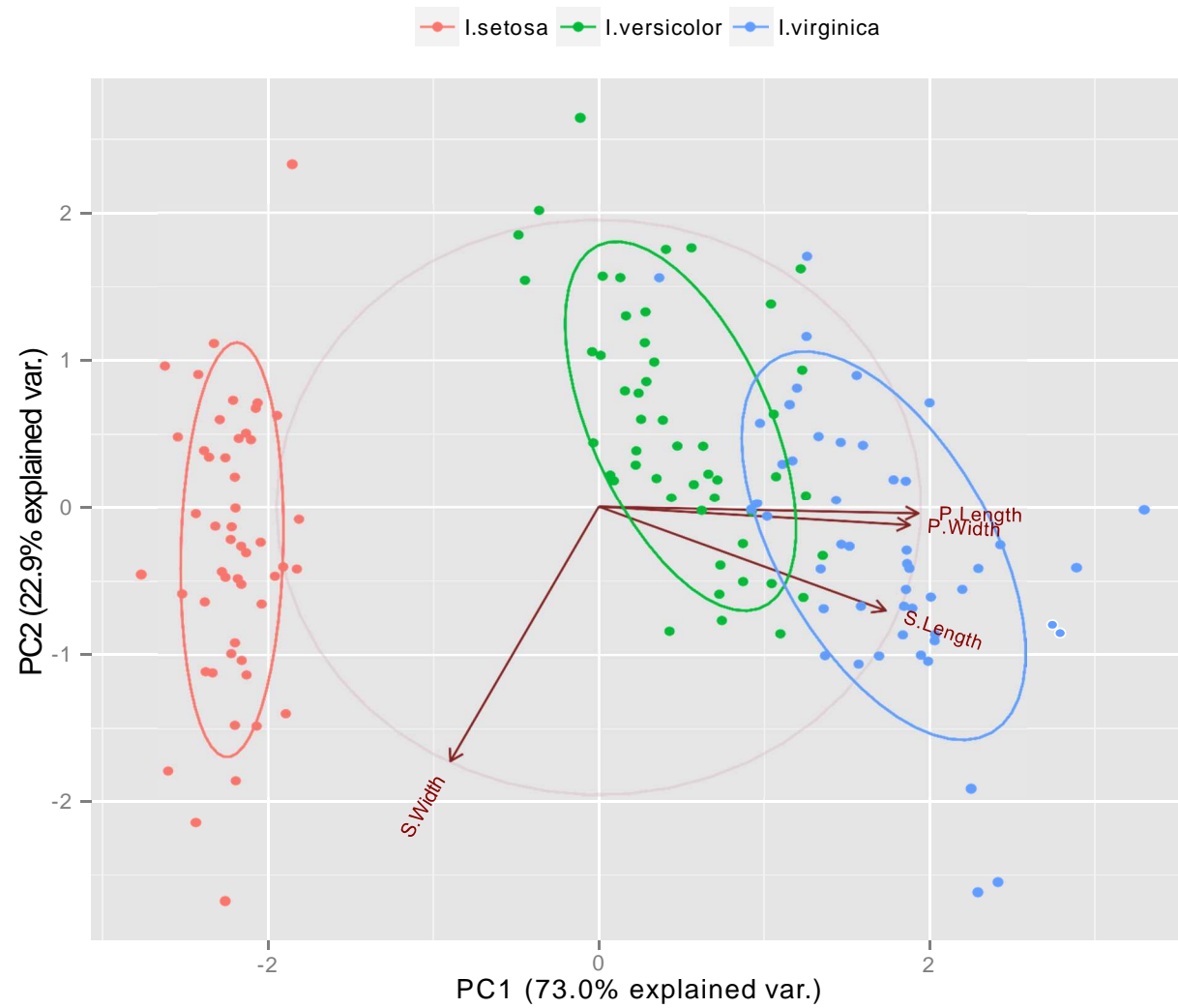
```
> summary(pca)
```

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

```
library(devtools)
install_github("vqv/ggbiplot")
library(ggbiplot)
ggbiplot(pca, obs.scale = 1, var.scale = 1, groups=iris
$Species, ellipse=T, circle=T) +
scale_color_discrete(name='') + theme(legend.direction =
'horizontal', legend.position="top")
```





# Tóm tắt

- PCA là một phương pháp giảm độ liên quan đa chiều của dữ liệu
- Ý tưởng: tìm các biến số mới (PC) là hàm số của các biến số gốc sao cho các PC không tương quan với nhau (orthogonal)
- PC đầu tiên giải thích nhiều phương sai nhất; PC 2 giải thích tỉ lệ phương sai ít hơn PC 1, v.v.

# Questions?

