



**UNSW**  
SYDNEY | Australia's  
Global  
University

**COMP9321**

# **Data Services Engineering**

**Term 1, 2019**

**Week 3 Lecture 2**

# Q5, Quiz 1

- The last option is also correct
- Everyone get 1 free mark capped to 6/6

```
CREATE TABLE CONTACTS(  
    ID INT PRIMARY KEY,  
    FIRST_NAME      CHAR(100),  
    LAST_NAME       CHAR(100),  
    PHONE_NUMBER    CHAR(32),  
    ADDRESS         CHAR(100),  
    POST_CODE       INT,  
);  
-----  
-----
```

- **nullable** – When set to `False`, will cause the “NOT NULL” phrase to be added when generating DDL for the column. When `True`, will normally generate nothing (in SQL this defaults to “NULL”), except in some very specific backend-specific edge cases where “NULL” may render explicitly. `Defaults to True unless primary_key is also True`, in which case it defaults to `False`. This parameter is only used when issuing CREATE TABLE statements.

# Assignment 1

Submission dryrun test is **disabled** as it raises confusions

- It is meant to check your format only
- Assignments will be reviewed by human

Assignment description is finalized last week

- Submission meet specifications will be marked correct
  - PDF and instructions, **excluding forum discussions**
- Sample output is for **format** demonstration only, we are not giving out answers
- Formatting is also important part of data engineering

# Data Visualization(2)

COMP9321 2019T1

# Iris Dataset

The measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each

of 3 species of iris →  $n = 150$

$d = 4$  (petal width, petal length, sepal length, sepal width)



**Iris Versicolor**

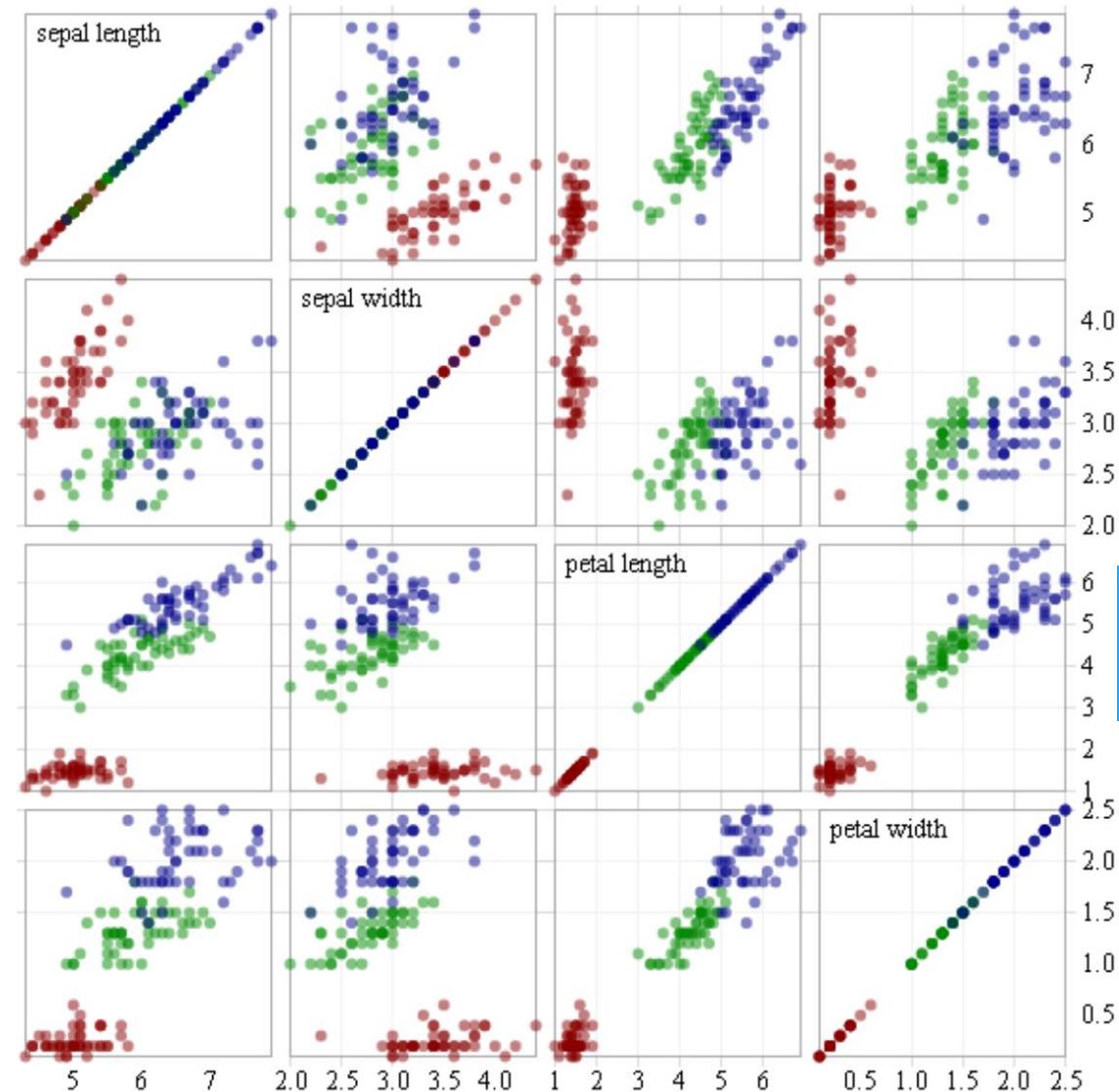


**Iris Setosa**



**Iris Virginica**

# How to Visualization?



How many do we need?  
 $2^d$

# Another Example: more dimensions

Given 53 blood and urine samples (features) from 65 people.

- How can we visualize the measurements?

Instances

|    | H-WBC  | H-RBC  | H-Hgb   | H-Hct   | H-MCV    | H-MCH   | H-MCHC  |
|----|--------|--------|---------|---------|----------|---------|---------|
| A1 | 8.0000 | 4.8200 | 14.1000 | 41.0000 | 85.0000  | 29.0000 | 34.0000 |
| A2 | 7.3000 | 5.0200 | 14.7000 | 43.0000 | 86.0000  | 29.0000 | 34.0000 |
| A3 | 4.3000 | 4.4800 | 14.1000 | 41.0000 | 91.0000  | 32.0000 | 35.0000 |
| A4 | 7.5000 | 4.4700 | 14.9000 | 45.0000 | 101.0000 | 33.0000 | 33.0000 |
| A5 | 7.3000 | 5.5200 | 15.4000 | 46.0000 | 84.0000  | 28.0000 | 33.0000 |
| A6 | 6.9000 | 4.8600 | 16.0000 | 47.0000 | 97.0000  | 33.0000 | 34.0000 |
| A7 | 7.8000 | 4.6800 | 14.7000 | 43.0000 | 92.0000  | 31.0000 | 34.0000 |
| A8 | 8.6000 | 4.8200 | 15.8000 | 42.0000 | 88.0000  | 33.0000 | 37.0000 |
| A9 | 5.1000 | 4.7100 | 14.0000 | 43.0000 | 92.0000  | 30.0000 | 32.0000 |

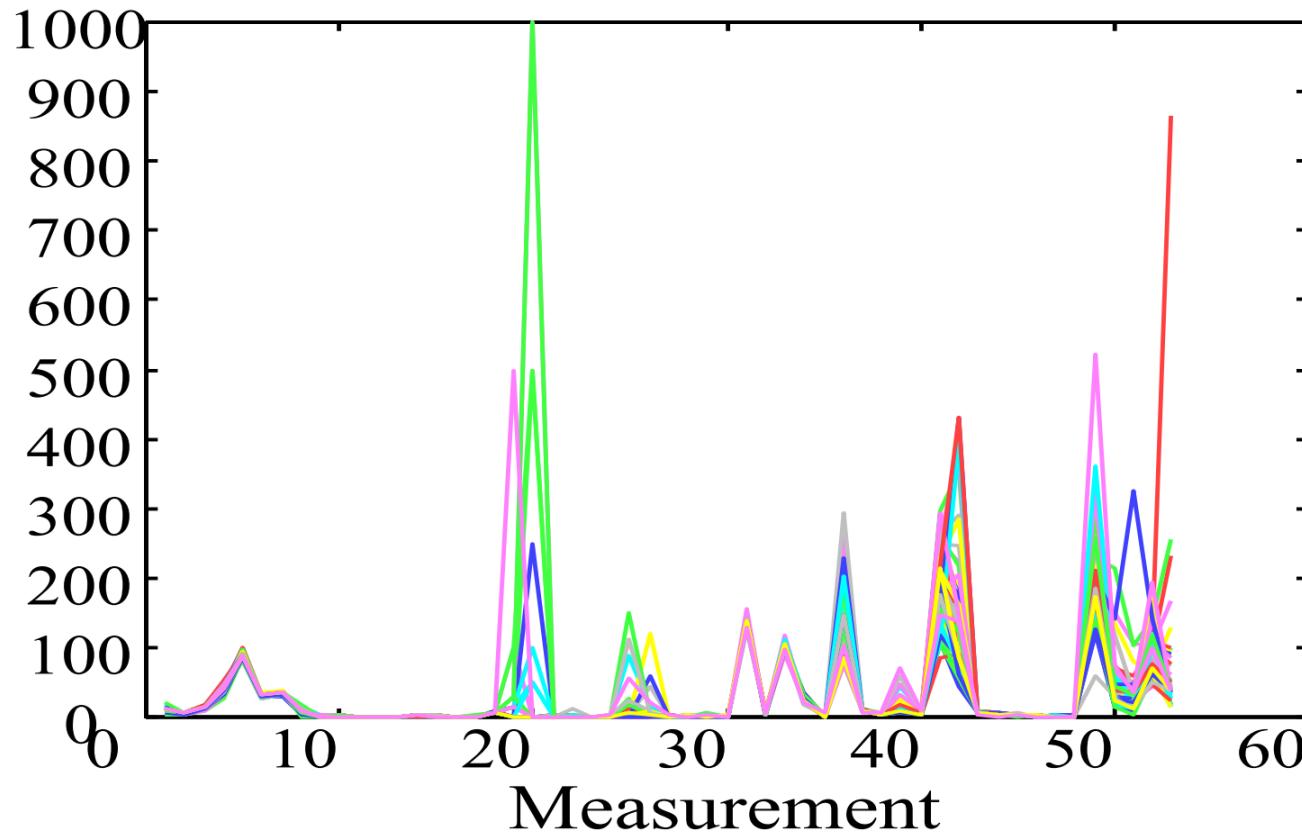
Features

# Another Example: more dimensions

Given 53 blood and urine samples (features) from 65 people.

- How can we visualize the measurements?
- 65 curves for 65 subjects !

Hard to compare between features

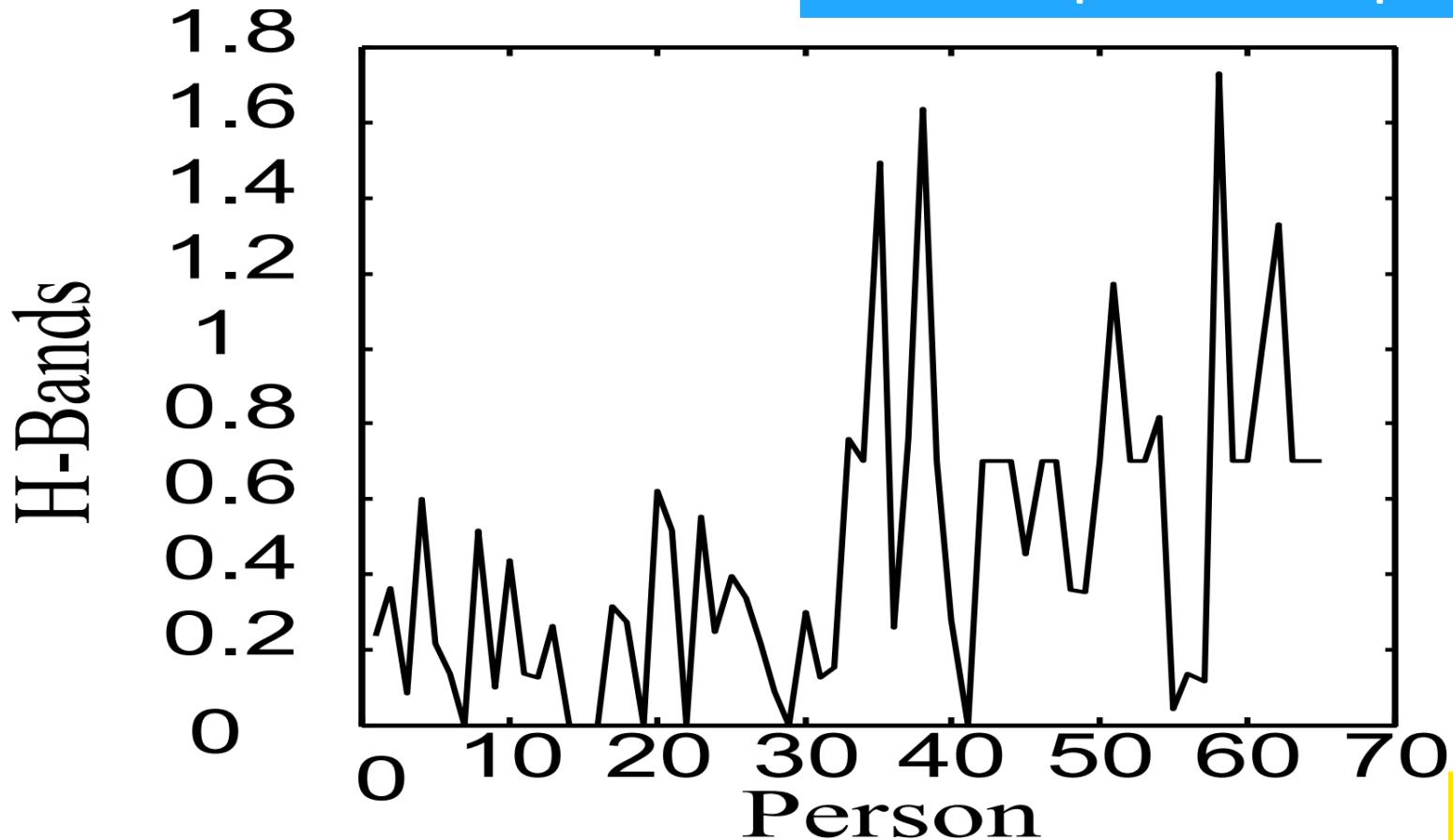


# Another Example: more dimensions

Given 53 blood and urine samples (features) from 65 people.

- How can we visualize the measurements?
- 53 pictures, one for each feature!

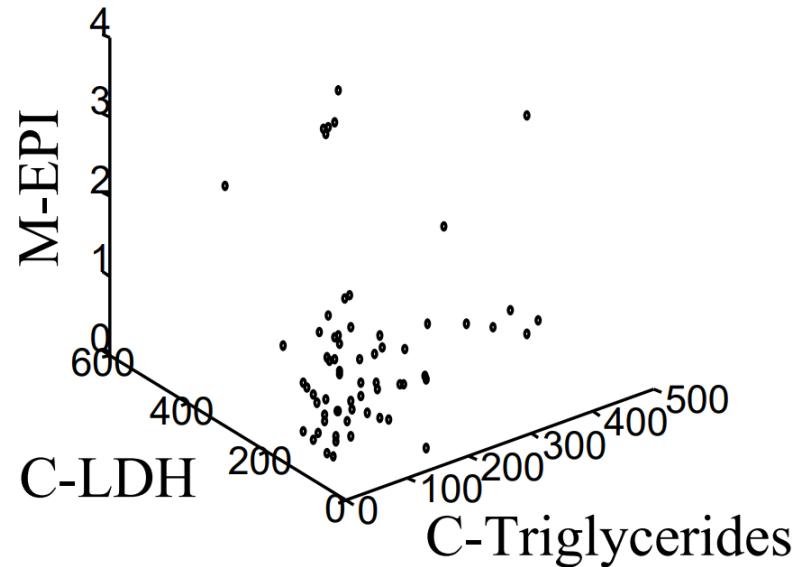
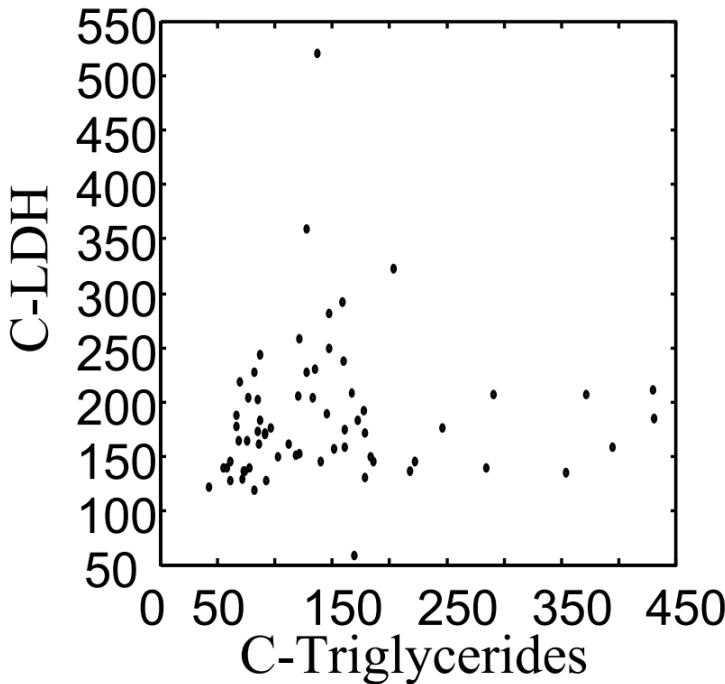
Hard to compare between patients



# Another Example: more dimensions

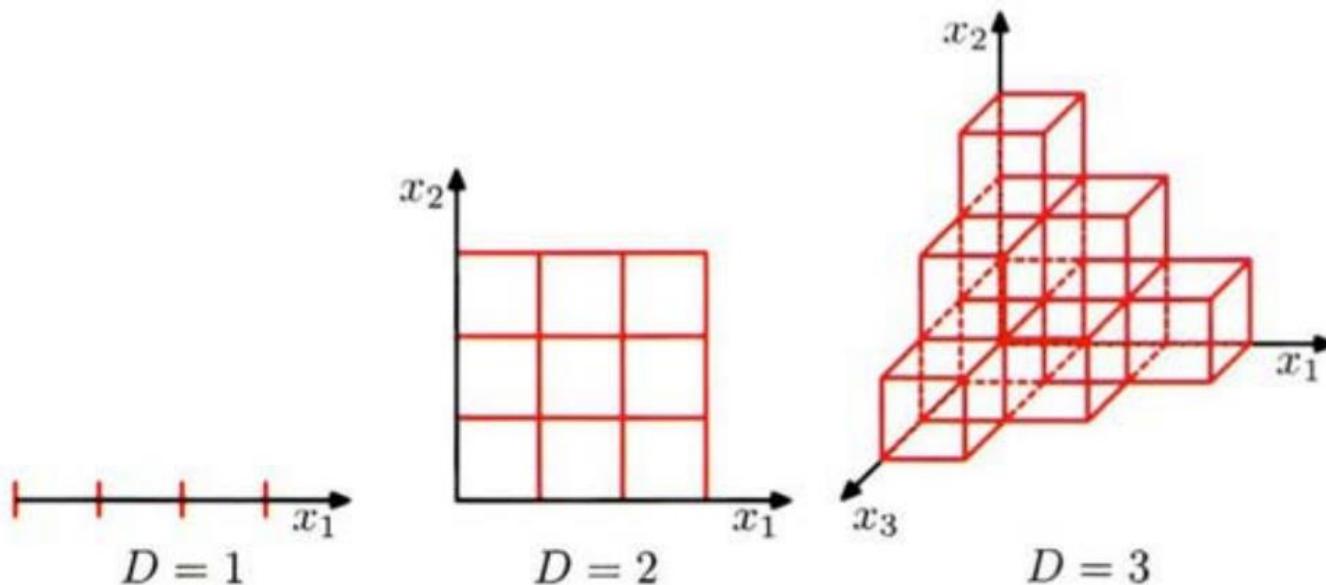
Given 53 blood and urine samples (features) from 65 people.

- How can we visualize the measurements?



# High Dimensional Data

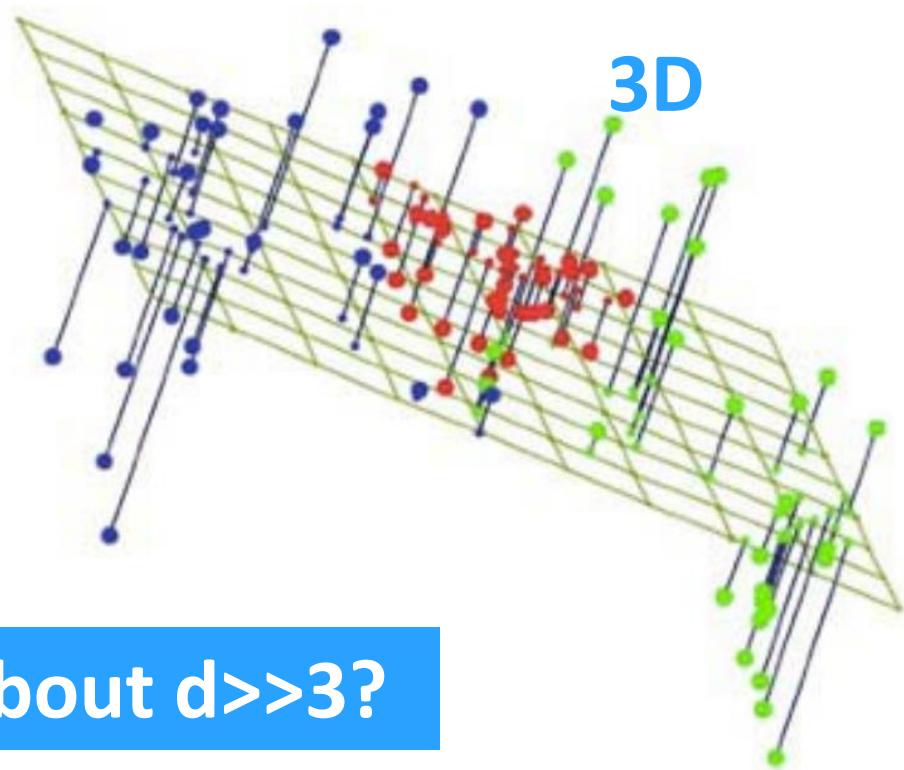
- What if the dimensionality increases to hundreds or thousands?
- Curse of dimensionality:
  - When dimensionality increases, the volume of the space increases so fast that the available data become sparse.
  - Statistically sound results require the sample size to grow exponentially with increasing dimensionality.



# Dimensionality Reduction

Project high-dimensional data to lower-dimensional subspace

- Linear projections
- Non-linear projections



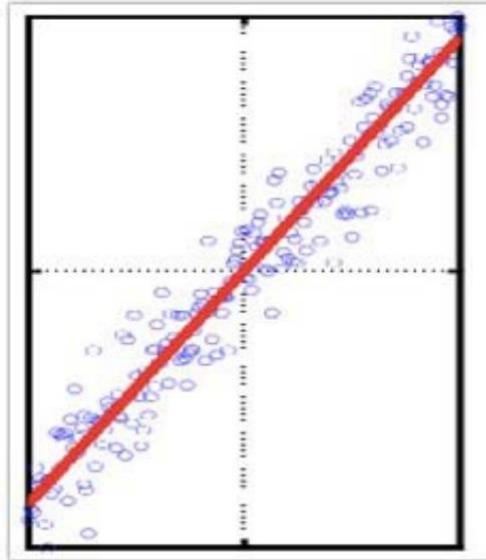
How about  $d \gg 3$ ?

# Dimensionality Reduction

- Does the data lie in / close to a hyperplane?
- What is the dimensionality of the hyperplane?

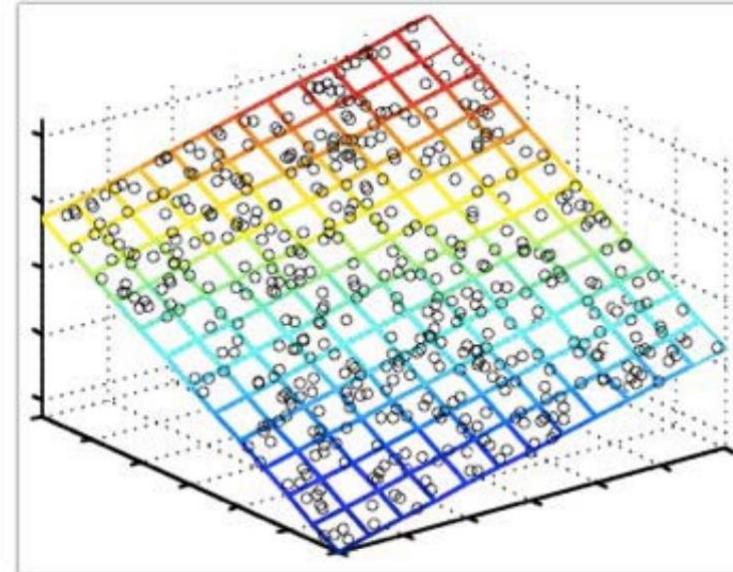
**D = 2**

**d = 1**



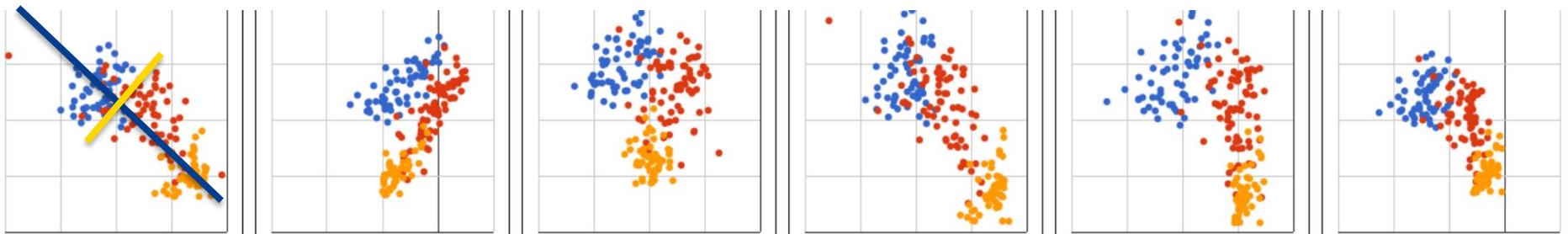
**D = 3**

**d = 2**



# Visualization using linear projections

- Consider a linear projection from a high-dimensional attribute space into a 2D (or 3D) visual space.
- Try to find an optimal projection with respect to some metric.
- They give the best 2D (or 3D) view on the data.
- It is a slice through the high-dimensional space.
- The visual encoding is that of a scatterplot.



# PCA

*The central idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set. This is achieved by transforming to a new set of variables, the principal components (PCs), which are uncorrelated, and which are ordered so that the first few retain most of the variation present in all of the original variables.*

[Jolliffe, Principal Component Analysis, 2nd edition]

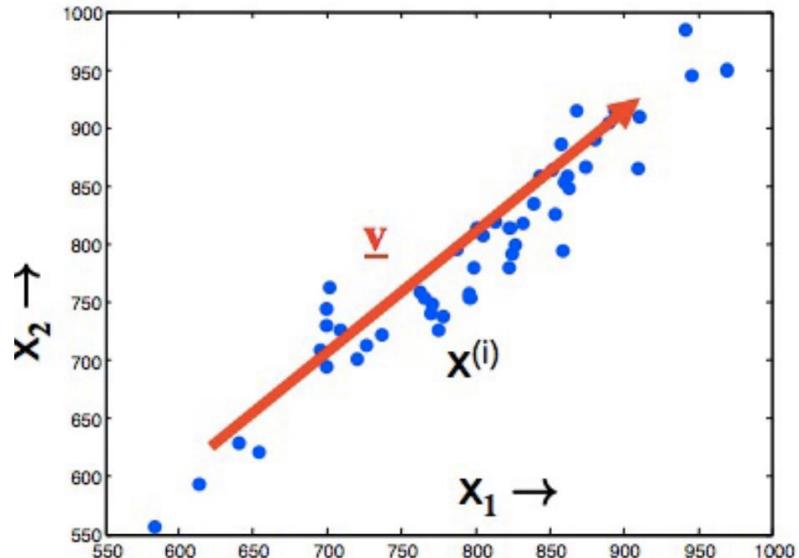
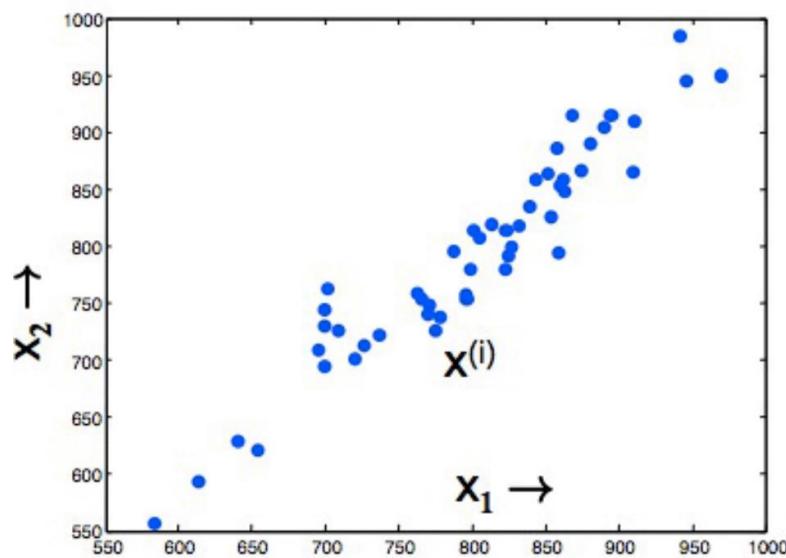
# Dimensionality Reduction

Idea:

- Given data points in d-dimensional space,
- Project into lower dimensional space while preserving as much information as possible
  - E.g., find best planar approximation to 3D data
  - E.g., find best planar approximation to 104D data
- In particular, choose projection that minimizes the squared error in reconstructing original data

# Principal Component Analysis

- PCA tries to find the most relevant directions (principal components) in the data.
- For projection to 2D, the first two principal components span the projection space



$$\vec{x} = [x_1, x_2]$$

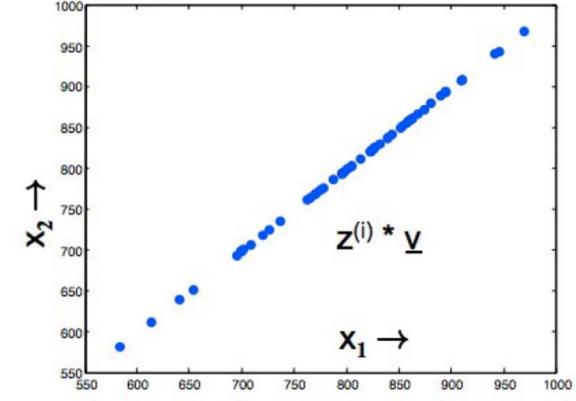
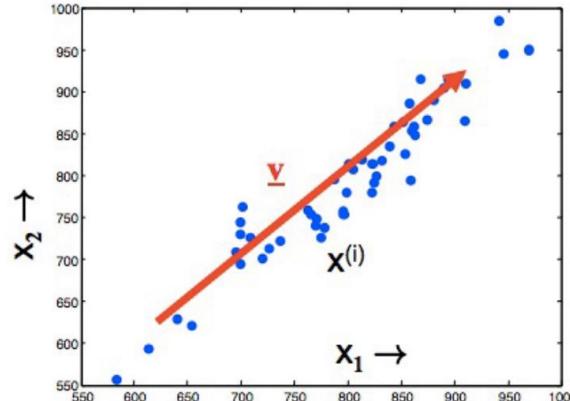
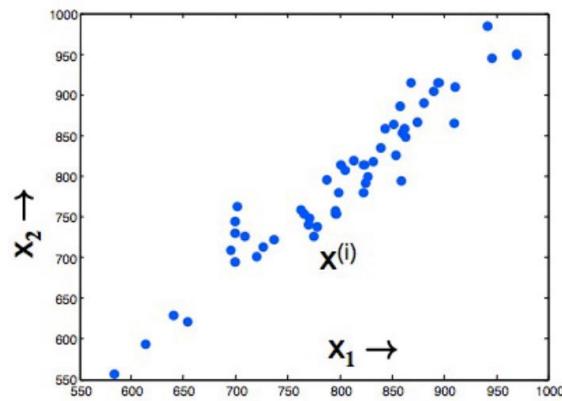
$$\vec{x} \approx s\vec{v} = s[v_1, v_2]$$

# PCA

1.  $\mathbf{v}$  is chosen to minimize residual variance

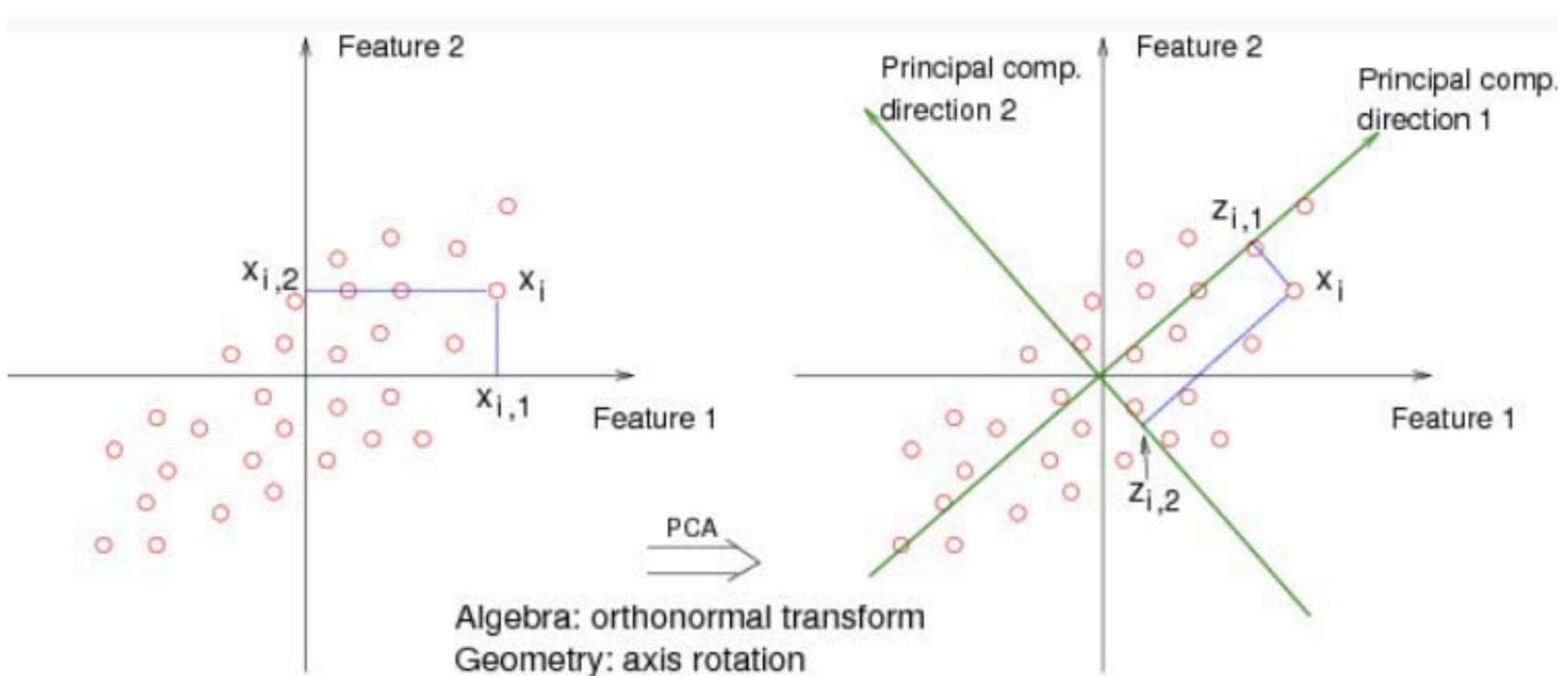
$$\min_{a,v} \sum_i (x^{(i)} - a^{(i)}v)^2$$

2. Find  $\mathbf{v}$  that most closely reconstructs  $\mathbf{x}$ .
3. Equivalent to  $\mathbf{v}$  being the direction of maximum variance



# PCA

Project data to subspace such as to maximize the variance of projected data



# PCA Process

- Input: Data  $X$  of sample size  $N$ .
- Output:  $k$  principal components
- Centering: Subtract mean from data
- Scaling: Scale each dimension by its variance
- Compute covariance matrix by  
$$\text{Cov}(\cdot) = \frac{1}{N} X^T X$$
- Compute  $k$  largest eigenvectors of  $\text{Cov}$

# PCA Algorithm – Formulation

## Basics of Linear Algebra

- Variance measures the spread of data in a dataset from the mean

$$var(X) = \frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n - 1}$$

- Covariance measures how each of the dimensions varies from the mean with respect to each other

$$cov(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n - 1}$$

# PCA Algorithm – Formulation

## Basics of Linear Algebra

- Positive covariance of two dimensions indicates that they change together (number of hours spent studying – grade)
- Negative covariance indicates that change in one dimension causes inverse change in the other (number of hours spent in a pub – balance of your bank account)
- Covariance matrix is a matrix of all pairwise covariances, e.g. for 3 dimensions X, Y, Z:

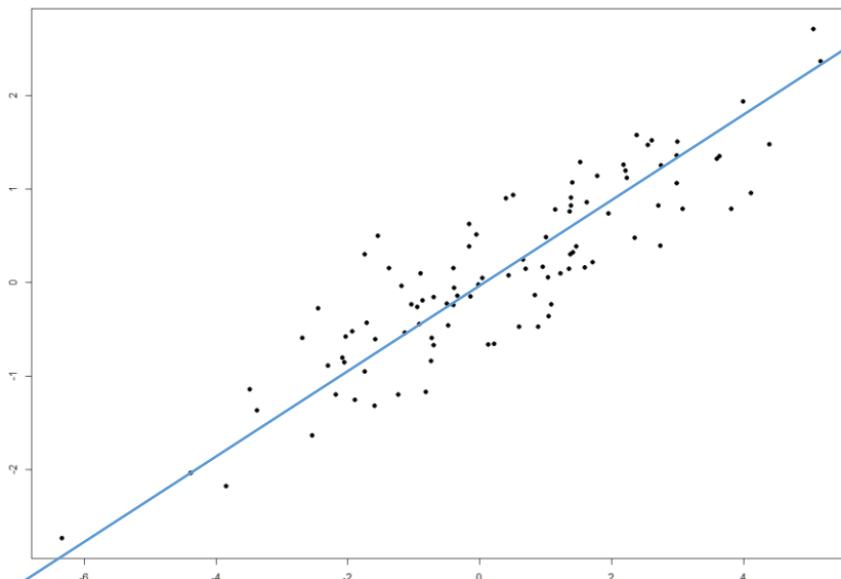
$$\begin{pmatrix} \text{cov}(X, X) & \text{cov}(X, Y) & \text{cov}(X, Z) \\ \text{cov}(Y, X) & \text{cov}(Y, Y) & \text{cov}(Y, Z) \\ \text{cov}(Z, X) & \text{cov}(Z, Y) & \text{cov}(Z, Z) \end{pmatrix}$$

# PCA Algorithm – Formulation

Let us have a random variable (observations)  $x^T = (x_1, \dots, x_p)$  with mean  $\mu$  and covariance matrix  $\Sigma$

First PC is the linear combination  $y_1 = a_1^T x = \sum_{i=1}^p a_{1i} x_i$

where  $a_1$  is chosen such that  $\text{var}(y_1)$  is maximum subject to  $a_1^T a_1 = 1$

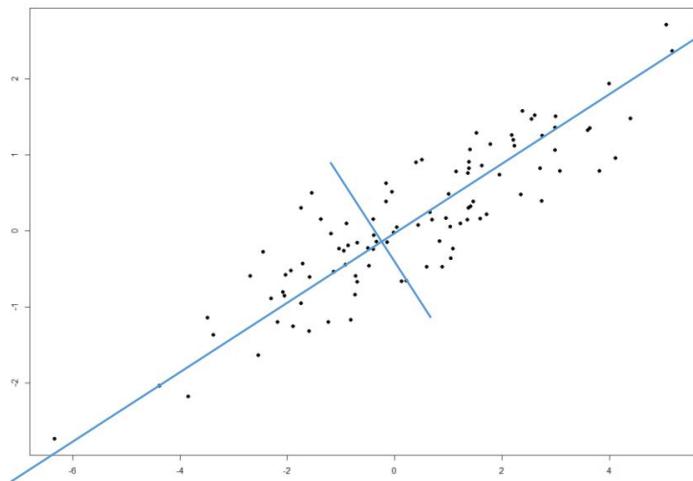


If we project the data onto this line, we lose as little information as possible = we keep as much variance as possible.

# PCA Algorithm – Formulation

Second PC is the linear combination

$$y_2 = a_2^T x = \sum_{i=1}^p a_{2i} x_i$$



where  $a_2$  is chosen such that  $\text{var}(y_2)$  is maximum  
subject to

$$a_2^T a_2 = 1 \text{ and } a_2^T a_1 = 0 = \text{cov}(a_k, a_l)$$

# PCA Algorithm – Formulation

Searching for the first PC

Assumption that the data are normalized, i.e., the mean is subtracted

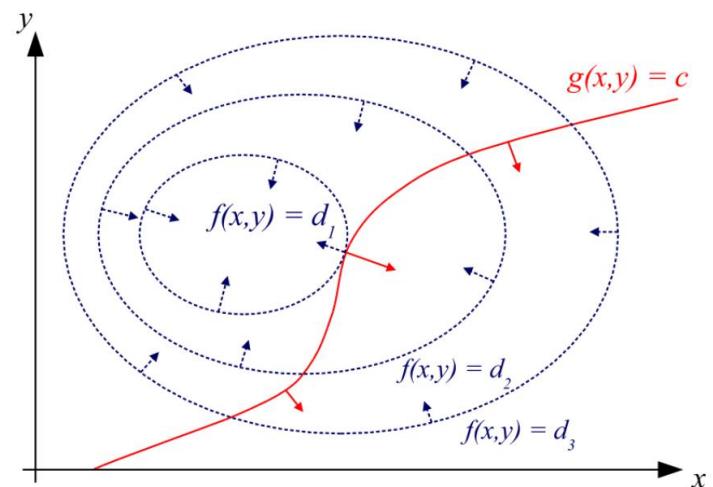
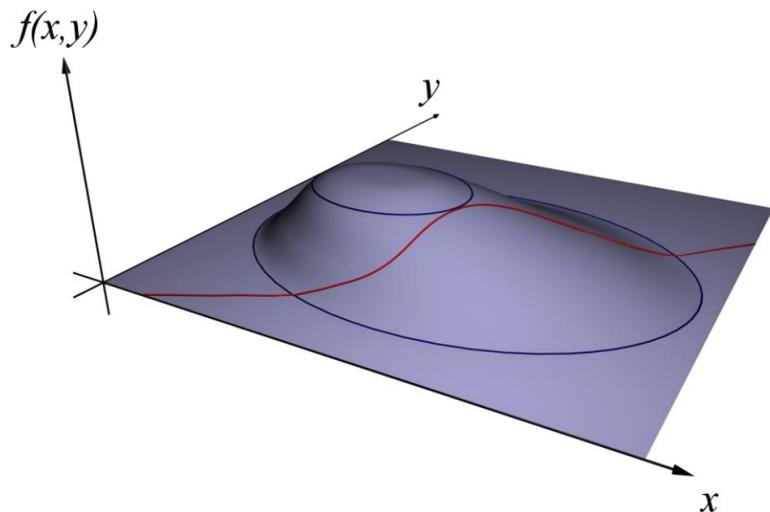
Find 1D subspace so that the observations have maximum spread in it  
→ maximizing variance

$$\begin{aligned} \text{var}(y_1) &= \text{var}(a_1^T X) = E[(a_1^T X - E[a_1^T X])(a_1^T X - E[a_1^T X])^T] \\ &= E[(a_1^T X)(a_1^T X)^T] = E[a_1^T X X^T a_1] = E[a_1^T \Sigma a_1] = a_1^T \Sigma a_1 \end{aligned}$$

The goal is to maximize variance given  $a_1^T a_1 = 1$

Lagrange multipliers

# Lagrange multipliers



source: Wikipedia

Maximize  $f(x, y)$  subject to  $g(x, y) = c \rightarrow$  introduction of a new variable - Lagrange multiplier  $\lambda$  ( $\nabla f = \lambda \nabla g \rightarrow \nabla f - \lambda \nabla g = 0$ )

$$\Lambda(x, y, \lambda) = f(x, y) + \lambda(g(x, y) - c) \rightarrow \frac{\Delta \Lambda(x, y, \lambda)}{\Delta x, y, \lambda} = 0$$

13

# PCA Algorithm – Formulation

- Transcription into the Lagrangian form

$$\Lambda(a_1, \lambda) = a_1^T \Sigma a_1 - \lambda(a_1^T a_1 - 1)$$

- Now we need to differentiate the Lagrangian

$$\frac{\partial \Lambda(a_1, \lambda)}{\partial a_1} = \frac{\partial \Lambda(a_1)}{\partial \begin{bmatrix} a_{11} \\ \dots \\ a_{1k} \end{bmatrix}} = 2\Sigma a_1 - 2\lambda a_1 = 0$$

# PCA Algorithm – Formulation

- This leads to the eigenproblem  $\Sigma \mathbf{a}_1 = \lambda \mathbf{a}_1 \rightarrow \mathbf{a}_1$  is an eigenvector of  $\lambda$

$$\text{var}(\mathbf{y}_1) = \text{var}(\mathbf{a}_1^T \mathbf{X}) = \mathbf{a}_1^T \Sigma \mathbf{a}_1 = \lambda \mathbf{a}_1^T \mathbf{a}_1 = \lambda$$

$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \rightarrow$  to maximize  $\text{var}(\mathbf{y}_1)$

$$\lambda = \lambda_1$$

# PCA Algorithm – Formulation

Searching for the next PCs

$$\begin{aligned}\Lambda(a_2) &= a_2^T \Sigma a_2 - \lambda(a_2^T a_2 - 1) - \kappa(a_2^T a_1) \\ \Sigma a_2 - \lambda a_2 - \kappa a_1 &= 0 \\ a_1^T \Sigma a_2 - \lambda a_1^T a_2 - \kappa a_1^T a_1 &= 0 \\ 0 - 0 - \kappa &= 0 \\ a_1^T \Sigma a_2 = a_2^T (\Sigma a_1) &= \lambda_1 a_2^T a_1 \\ \Sigma a_2 - \lambda a_2 &= 0 \\ \Sigma a_2 = \lambda a_2 &\Rightarrow \lambda = \lambda_2\end{aligned}$$

# PCA Algorithm – Formulation

Thus the coefficients of the linear combination which transform the observations onto the PCs are formed by eigenvalues of the covariance matrix

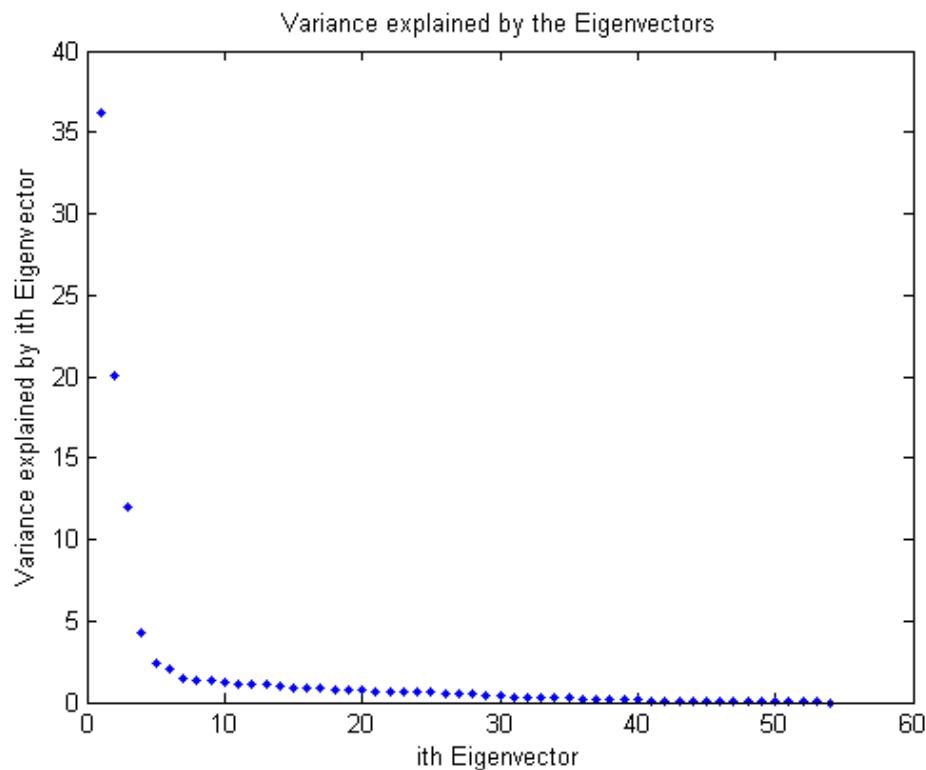
Let  $A$  contain the eigenvalues  $a_i$  as its columns and let  $x$  be a  $p$ -dimensional vector, then

$$\mathbf{y} = \mathbf{A}^T(\mathbf{x} - \boldsymbol{\mu})$$

# How many components do I need?

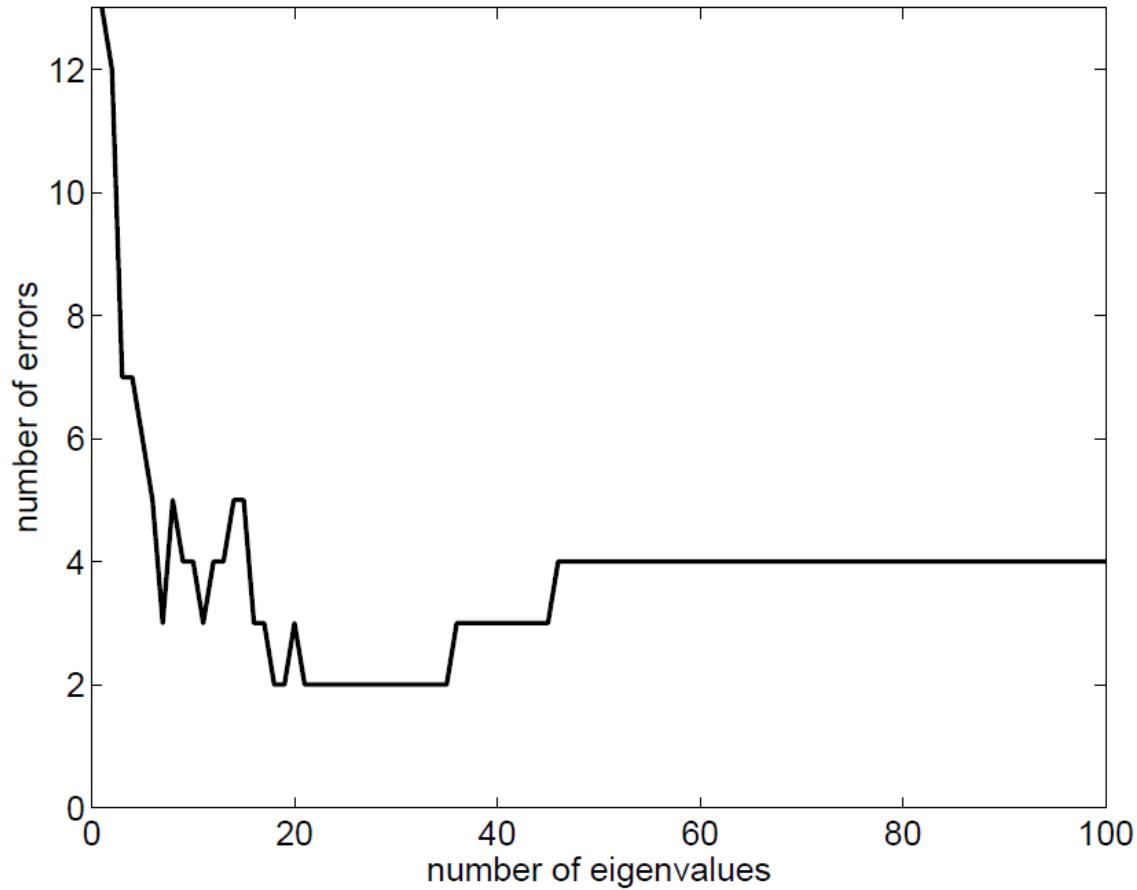
Check the distribution of eigen-values

Take enough many eigen-vectors to cover 80-90% of the variance



# How many components do I need?

Check the validation errors



# A 2D Numerical Example

# PCA Example –STEP 1

## Subtract the mean

from each of the data dimensions. All the x values have x subtracted and y values have y subtracted from them. This produces a data set whose  $\bar{m}$ ean is zero.

Subtracting the mean makes variance and covariance calculation easier by simplifying their equations. The variance and co-variance values are not affected by the mean value.

# PCA Example –STEP 1

<http://kybele.psych.cornell.edu/~edelman/Psych-465-Spring-2003/PCA-tutorial.pdf>

DATA:

| x   | y   |
|-----|-----|
| 2.5 | 2.4 |
| 0.5 | 0.7 |
| 2.2 | 2.9 |
| 1.9 | 2.2 |
| 3.1 | 3.0 |
| 2.3 | 2.7 |
| 2   | 1.6 |
| 1   | 1.1 |
| 1.5 | 1.6 |
| 1.1 | 0.9 |

ZERO MEAN DATA:

| x     | y     |
|-------|-------|
| .69   | .49   |
| -1.31 | -1.21 |
| .39   | .99   |
| .09   | .29   |
| 1.29  | 1.09  |
| .49   | .79   |
| .19   | -.31  |
| -.81  | -.81  |
| -.31  | -.31  |
| -.71  | -1.01 |

# PCA Example –STEP 1

<http://kybele.psych.cornell.edu/~edelman/Psych-465-Spring-2003/PCA-tutorial.pdf>

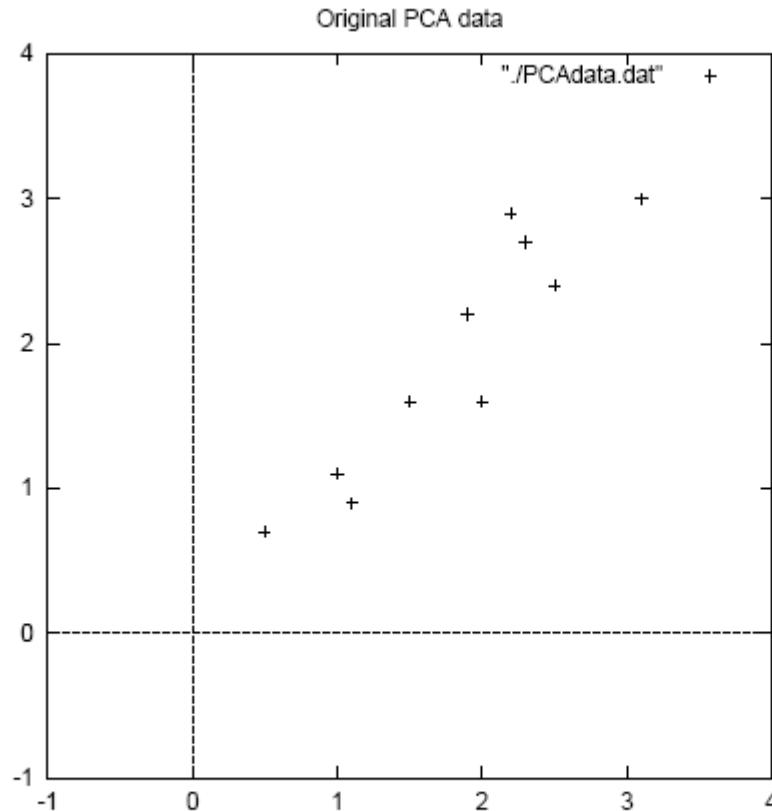


Figure 3.1: PCA example data, original data on the left, data with the means subtracted on the right, and a plot of the data

# PCA Example –STEP 2

Calculate the covariance matrix

$$\text{cov} = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

since the non-diagonal elements in this covariance matrix are positive, we should expect that both the x and y variable increase together.

# PCA Example –STEP 3

Calculate the eigenvectors and eigenvalues of the covariance matrix

$$\text{eigenvalues} = \begin{cases} .0490833989 \\ 1.28402771 \end{cases}$$

$$\text{eigenvectors} = \begin{cases} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{cases}$$

# PCA Example –STEP 3

<http://kybele.psych.cornell.edu/~edelman/Psych-465-Spring-2003/PCA-tutorial.pdf>

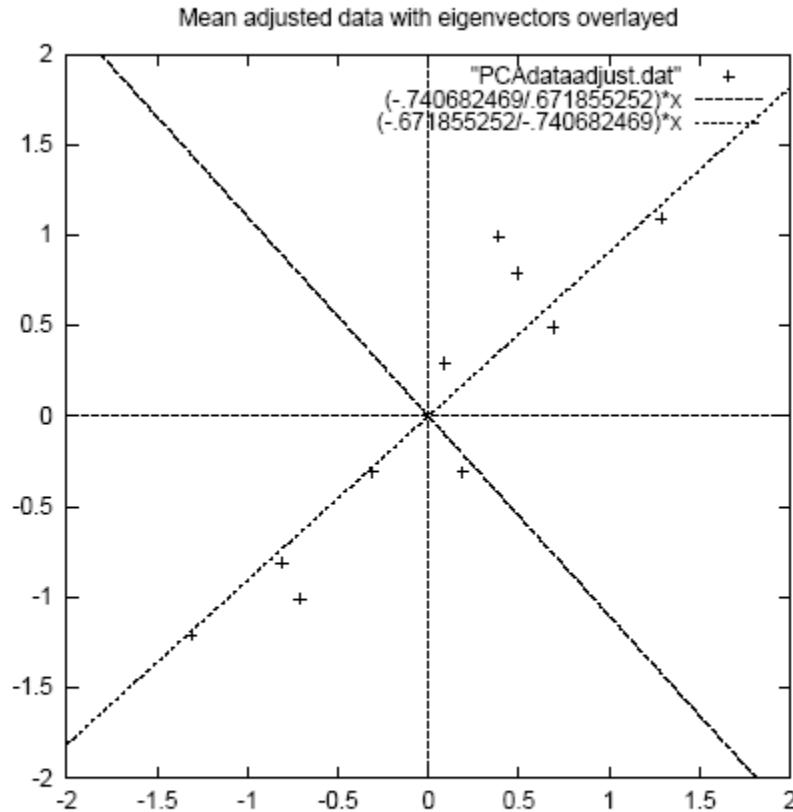


Figure 3.2: A plot of the normalised data (mean subtracted) with the eigenvectors of the covariance matrix overlayed on top.

- eigenvectors are plotted as diagonal dotted lines on the plot.
- Note they are perpendicular to each other.
- Note one of the eigenvectors goes through the middle of the points, like drawing a line of best fit.
- The second eigenvector gives us the other, less important, pattern in the data, that all the points follow the main line, but are off to the side of the main line by some amount.

# PCA Example –STEP 4

Now, if you like, you can decide to *ignore* the components of lesser significance.

You do lose some information, but if the eigenvalues are small, you don't lose much

n dimensions in your data  
calculate n eigenvectors and eigenvalues  
choose only the first p eigenvectors  
final data set has only p dimensions.

# PCA Example –STEP 4

## Feature Vector

FeatureVector = (eig<sub>1</sub> eig<sub>2</sub> eig<sub>3</sub> ... eig<sub>n</sub>)

We can either form a feature vector with both of the eigenvectors:

$$\begin{pmatrix} -.677873399 & -.735178656 \\ -.735178656 & .677873399 \end{pmatrix}$$

or, we can choose to leave out the smaller, less significant component and only have a single column:

$$\begin{pmatrix} -.677873399 \\ -.735178656 \end{pmatrix}$$

# PCA Example –STEP 5

## Deriving the new data

**FinalData = RowFeatureVector x RowZeroMeanData**

**RowFeatureVector** is the matrix with the eigenvectors in the columns *transposed* so that the eigenvectors are now in the rows, with the most significant eigenvector at the top

**RowZeroMeanData** is the mean-adjusted data *transposed*, ie. the data items are in each column, with each row holding a separate dimension.

# PCA Example –STEP 5

FinalData transpose: dimensions along columns

| x           | y           |
|-------------|-------------|
| -.827970186 | -.175115307 |
| 1.77758033  | .142857227  |
| -.992197494 | .384374989  |
| -.274210416 | .130417207  |
| -1.67580142 | -.209498461 |
| -.912949103 | .175282444  |
| .0991094375 | -.349824698 |
| 1.14457216  | .0464172582 |
| .438046137  | .0177646297 |
| 1.22382056  | -.162675287 |

# PCA Example –STEP 5

<http://kybele.psych.cornell.edu/~edelman/Psych-465-Spring-2003/PCA-tutorial.pdf>

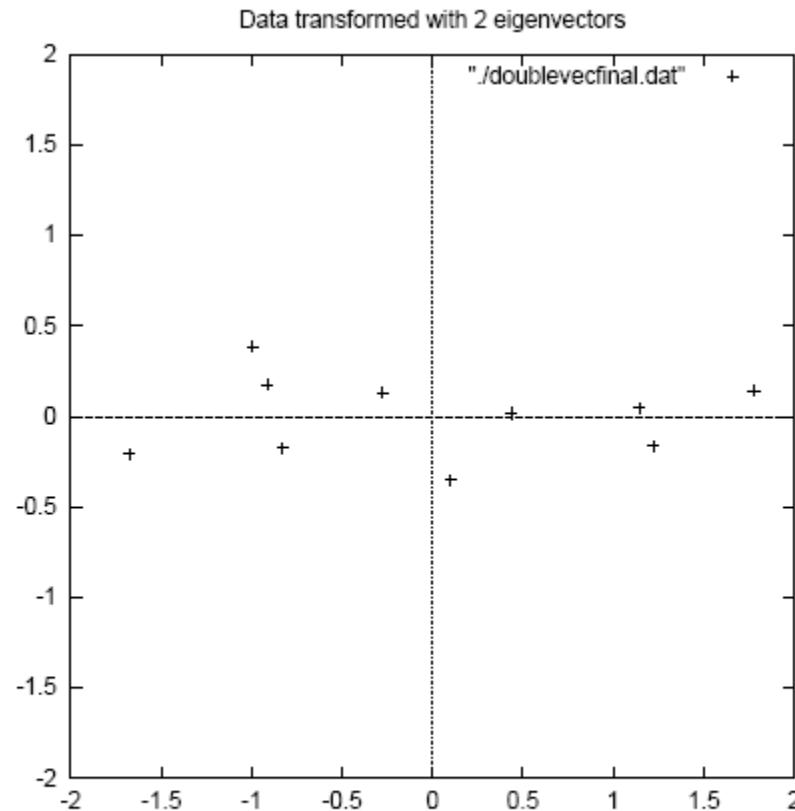


Figure 3.3: The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points.

# Reconstruction of original Data

If we reduced the dimensionality, obviously, when reconstructing the data we would lose those dimensions we chose to discard. In our example let us assume that we considered only the x dimension...

# Reconstruction of original Data

<http://kybele.psych.cornell.edu/~edelman/Psych-465-Spring-2003/PCA-tutorial.pdf>

X

-.827970186  
1.77758033  
-.992197494  
-.274210416  
-1.67580142  
-.912949103  
.0991094375  
1.14457216  
.438046137  
1.22382056

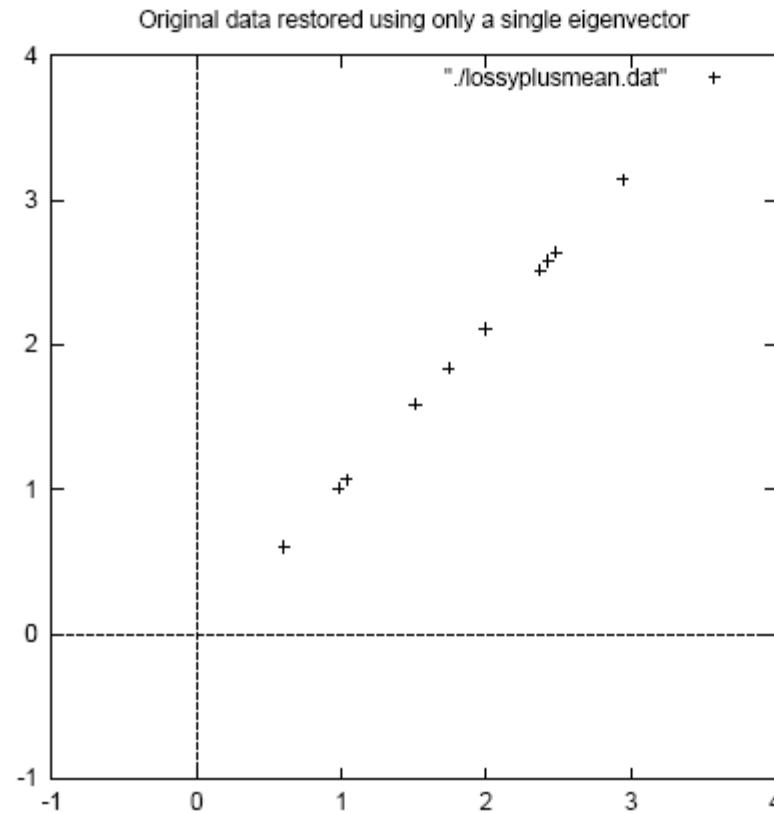


Figure 3.5: The reconstruction from the data that was derived using only a single eigenvector

# Problems and limitations

What if very large dimensional data?

- e.g., Images ( $d \geq 10^4$ )

Problem:

- Covariance matrix  $\Sigma$  is size  $(d^2)$ . The computational complexity is  $O(d^3)$ !
- $d=10^4 \rightarrow |\Sigma| = 10^4 \times 10^4 = 10^8 !!!$

May bound to Min  $O(D, N)^3$

## Singular Value Decomposition (SVD)!

- efficient algorithms available (Matlab)
- some implementations find just top  $N$  eigenvectors

# SVD

Singular Value Decomposition

# Singular Value Decomposition

Problem:

- #1: Find concepts in text
- #2: Reduce dimensionality

| term<br>document | data | information | retrieval | brain | lung |
|------------------|------|-------------|-----------|-------|------|
| CS-TR1           | 1    | 1           | 1         | 0     | 0    |
| CS-TR2           | 2    | 2           | 2         | 0     | 0    |
| CS-TR3           | 1    | 1           | 1         | 0     | 0    |
| CS-TR4           | 5    | 5           | 5         | 0     | 0    |
| MED-TR1          | 0    | 0           | 0         | 2     | 2    |
| MED-TR2          | 0    | 0           | 0         | 3     | 3    |
| MED-TR3          | 0    | 0           | 0         | 1     | 1    |

# SVD - Definition

$$\mathbf{A}_{[n \times m]} = \mathbf{U}_{[n \times r]} \Lambda_{[r \times r]} (\mathbf{V}_{[m \times r]})^T$$

**A**:  $n \times m$  matrix (e.g., n documents, m terms)

**U**:  $n \times r$  matrix (n documents, r concepts)

**$\Lambda$** :  $r \times r$  diagonal matrix (strength of each ‘concept’) (r: rank of the matrix)

**V**:  $m \times r$  matrix (m terms, r concepts)

# SVD - Properties

**THEOREM** [Press+92]: always possible to decompose matrix  $\mathbf{A}$  into  $\mathbf{A} = \mathbf{U} \Lambda \mathbf{V}^T$ , where

$\mathbf{U}, \Lambda, \mathbf{V}$ : unique (\*)

$\mathbf{U}, \mathbf{V}$ : column orthonormal (ie., columns are unit vectors, orthogonal to each other)

- $\mathbf{U}^T \mathbf{U} = \mathbf{I}; \mathbf{V}^T \mathbf{V} = \mathbf{I}$  ( $\mathbf{I}$ : identity matrix)

$\Lambda$ : singular value are positive, and sorted in decreasing order

# SVD - Properties

'spectral decomposition' of the matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} | & & | \\ u_1 & & u_2 \\ | & & | \end{bmatrix} \times \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \times \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

# SVD - Interpretation

‘documents’, ‘terms’ and ‘concepts’:

**U**: document-to-concept similarity matrix

**V**: term-to-concept similarity matrix

**Λ**: its diagonal elements: ‘strength’ of each concept

Projection:

best axis to project on: (‘best’ = min sum of squares of projection errors)

# SVD - Example

$\mathbf{A} = \mathbf{U} \Lambda \mathbf{V}^T$  - example:

↑  
CS  
↓  
MD

↑  
data  
↓  
inf. ↓ brain lung

↑  
retrieval

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

# SVD - Example

$A = U \Lambda V^T$  - example:

doc-to-concept  
similarity matrix

**CS-concept**      **MD-concept**

$$\begin{matrix} & \text{retrieval} & \text{inf.} \\ \text{data} & \downarrow & \text{brain} & \text{lung} \end{matrix}$$
$$\begin{matrix} \uparrow & \text{CS} \\ \downarrow & \\ \uparrow & \text{MD} \\ \downarrow & \end{matrix}$$
$$= \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

The matrix multiplication shows the decomposition of the document-term matrix (CS) into three components: CS-concept, MD-concept, and doc-to-concept similarity matrix. The circled value 0.18 in the CS-concept matrix indicates the similarity between the first document and the first concept.

# SVD - Example

## $A = U \Lambda V^T$ - example:

$$\begin{array}{c}
 \text{retrieval} \\
 \text{inf.} \downarrow \\
 \text{data} \quad \text{brain} \quad \text{lung}
 \end{array}
 \quad \text{'strength' of CS-concept}$$

↑ CS      ↓ MD

$$\begin{bmatrix}
 1 & 1 & 1 & 0 & 0 \\
 2 & 2 & 2 & 0 & 0 \\
 1 & 1 & 1 & 0 & 0 \\
 5 & 5 & 5 & 0 & 0 \\
 0 & 0 & 0 & 2 & 2 \\
 0 & 0 & 0 & 3 & 3 \\
 0 & 0 & 0 & 1 & 1
 \end{bmatrix}
 = \begin{bmatrix}
 0.18 & 0 \\
 0.36 & 0 \\
 0.18 & 0 \\
 0.90 & 0 \\
 0 & 0.53 \\
 0 & 0.80 \\
 0 & 0.27
 \end{bmatrix}
 \times \begin{bmatrix}
 9.64 & 0 \\
 0 & 5.29
 \end{bmatrix}
 \times \begin{bmatrix}
 0.58 & 0.58 & 0.58 & 0 & 0 \\
 0 & 0 & 0 & 0.71 & 0.71
 \end{bmatrix}$$

# SVD - Example

$A = U \Lambda V^T$  - example:

retrieval

inf. ↓      brain      lung

↑ CS

↓ MD

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

term-to-concept similarity matrix

CS-concept

Diagram illustrating the SVD decomposition of a document-term matrix  $A$  into three components:

- CS (Concept-Similarity) Matrix:** Represented by the first column of the matrix. It has columns labeled "retrieval", "inf.", "brain", "lung", and "MD".
- CS-concept Matrix:** Represented by the second matrix in the equation. It is a term-to-concept similarity matrix.
- MD (Matrix Decomposition) Matrix:** Represented by the third matrix in the equation. It is a diagonal matrix of singular values.

The circled value **0.58** in the CS-concept matrix corresponds to the value **0.58** in the MD matrix.

# SVD – Dimensionality reduction

Q: how exactly is dim. reduction done?

A: set the smallest singular values to zero:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix} \times \begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}$$

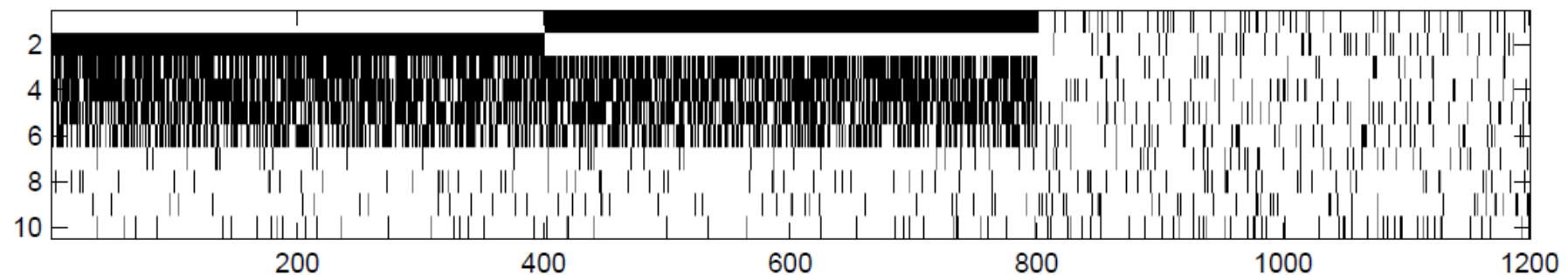
The diagram illustrates the Singular Value Decomposition (SVD) of a 7x5 matrix. The matrix is shown as a product of three matrices: U (left), S (diagonal), and V (right). The S matrix has 7 rows and 5 columns. Blue lines connect the non-zero entries in the S matrix to the corresponding columns in the U and V matrices. The diagonal entries of the S matrix are 0.18, 0.36, 0.18, 0.90, 0, 0.53, and 0.27. The V matrix has 5 columns, and its last two columns (labeled 4 and 5) are crossed out with a large blue X, indicating they are removed during dimensionality reduction. The U matrix has 7 rows and 5 columns, and the V matrix has 5 rows and 5 columns.

# SVD - Dimensionality reduction

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 0.18 \\ 0.36 \\ 0.18 \\ 0.90 \\ 0 \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} 9.64 \\ \vdots \end{bmatrix} \times \begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \end{bmatrix}$$

# SVD - Dimensionality reduction

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



# What Can PCA Do?

## Visualization

- Objects represented by many descriptors
- PCA helps to find structure among objects which could not be visualized otherwise (e.g., patients or car accidents)

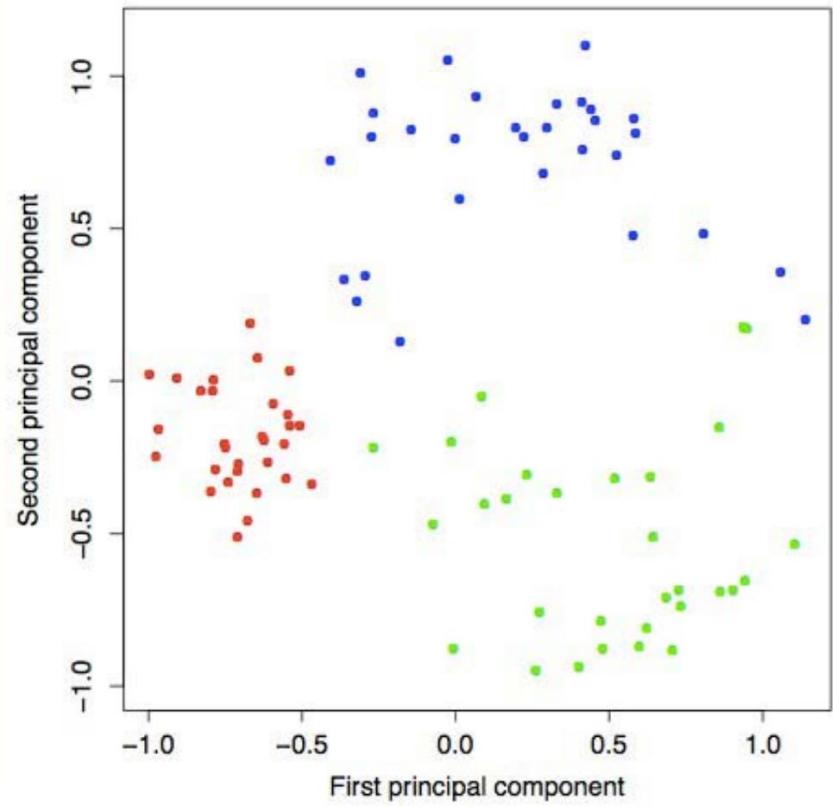
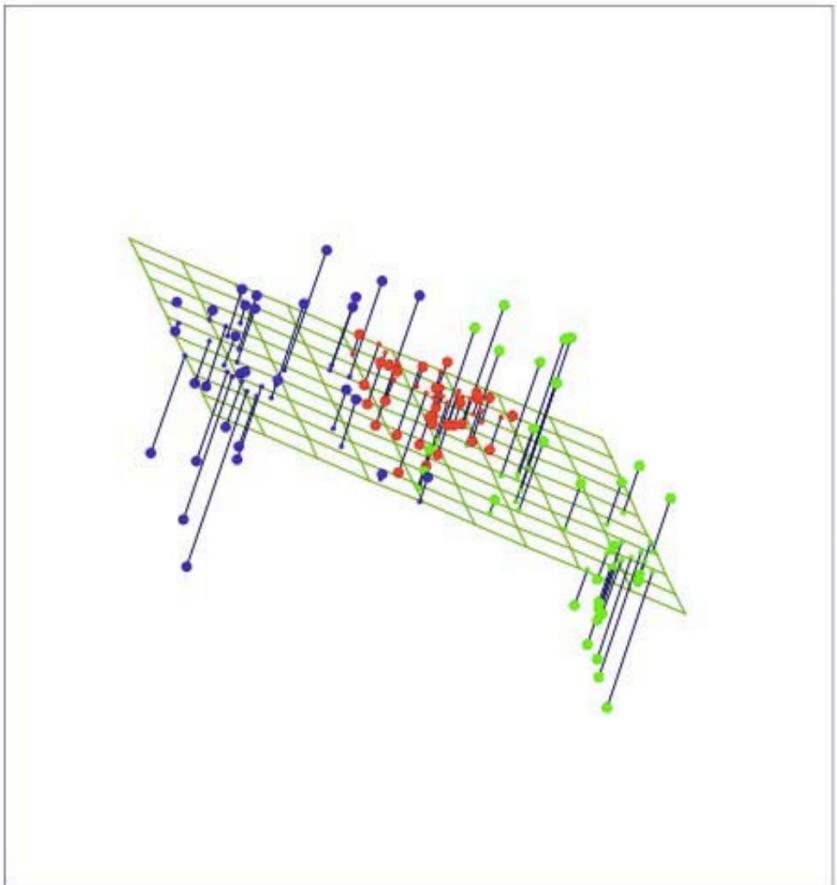
## Compression

- Representation of object only by their coordinates in the respective subspace
- E.g. in the eigenfaces (see later), each faces can be reasonable approximated by 10 coordinates

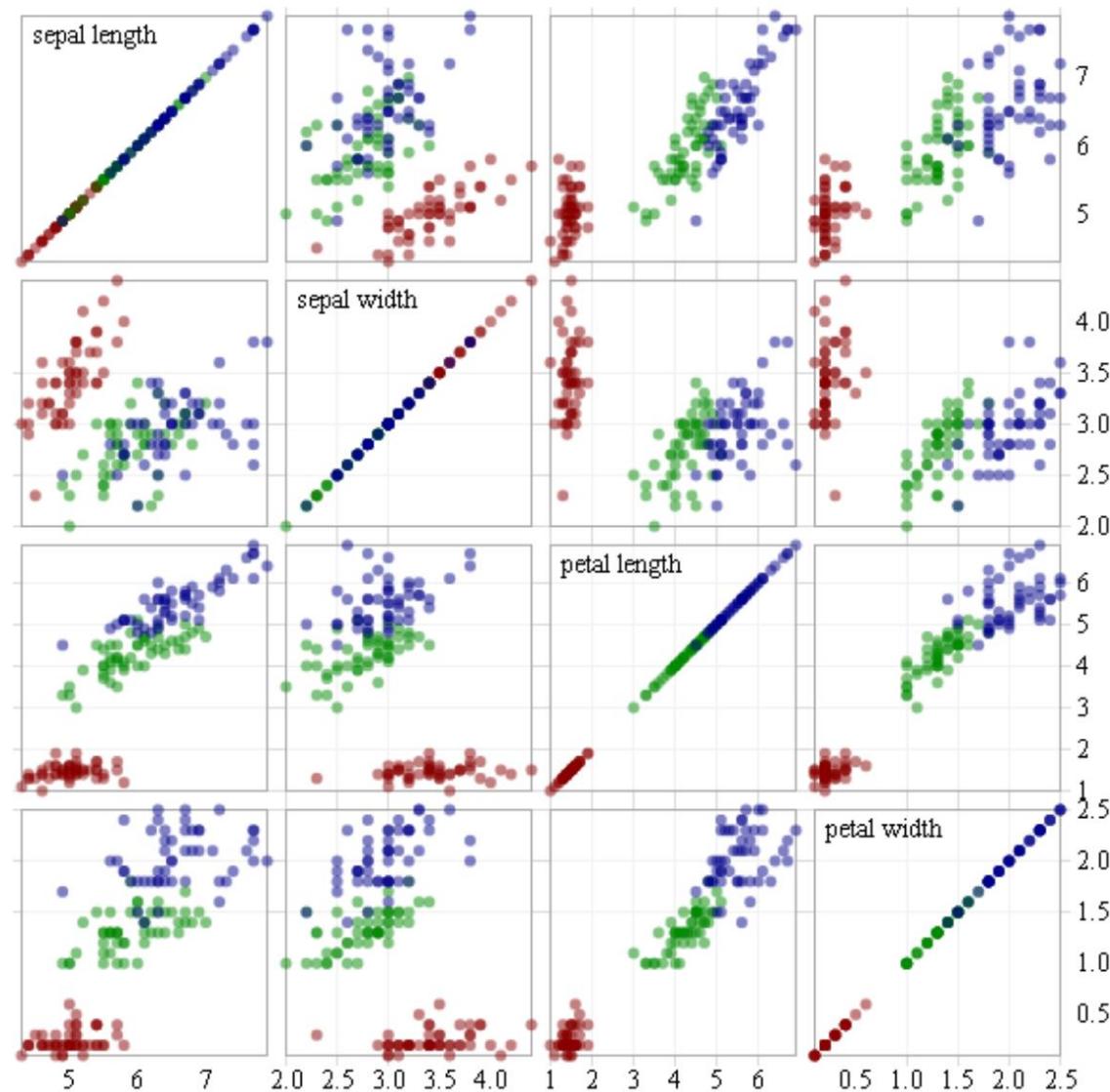
## Machine learning

- Dimension reduction pre-step
- Denoising

# What Can PCA Do?

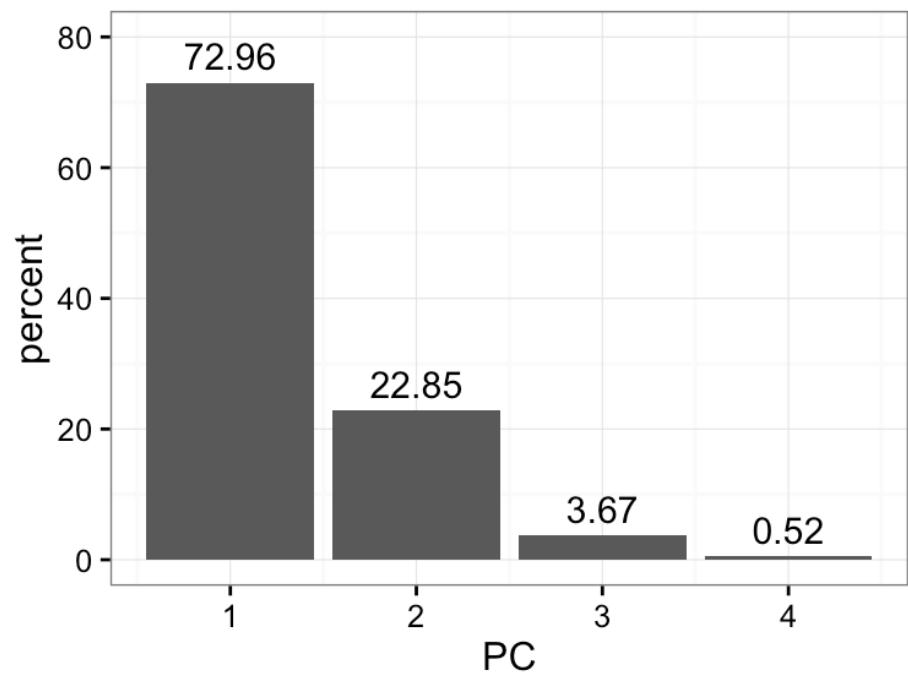
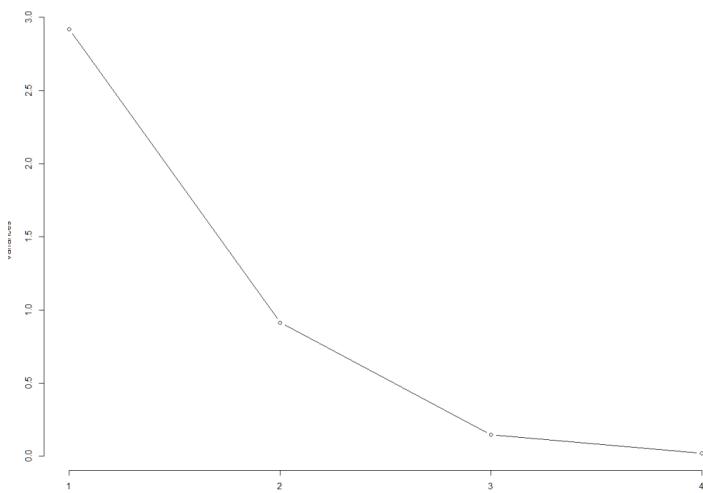


## Revisit Iris

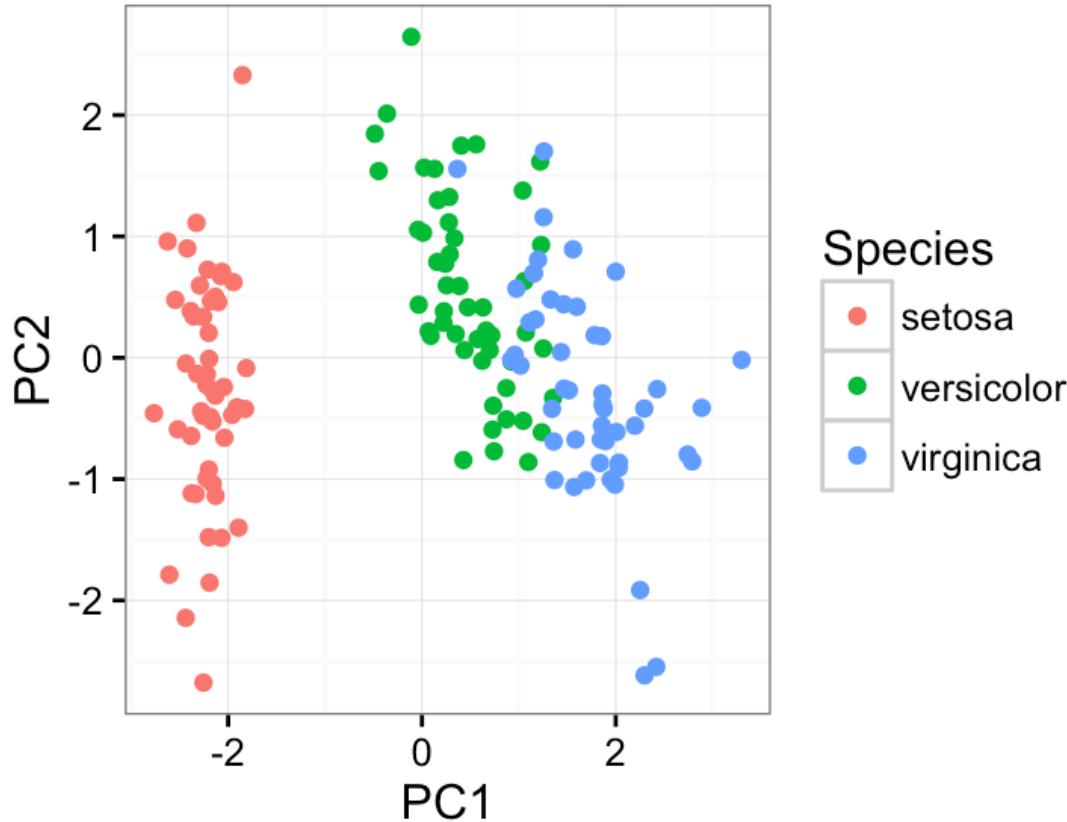


## Revisit Iris

- Display of variance of each of the component
- Plot of magnitudes of eigenvalues
- Gives impression of the intrinsic dimensionality

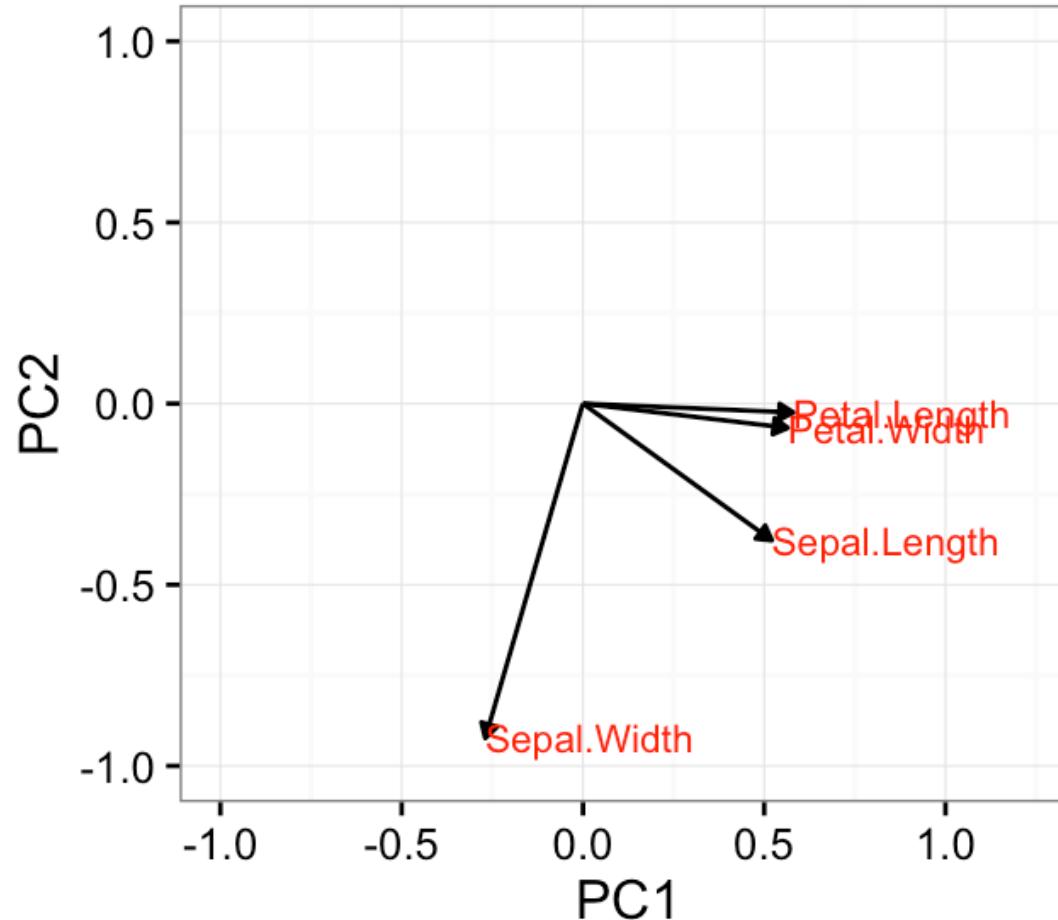


## Revisit Iris



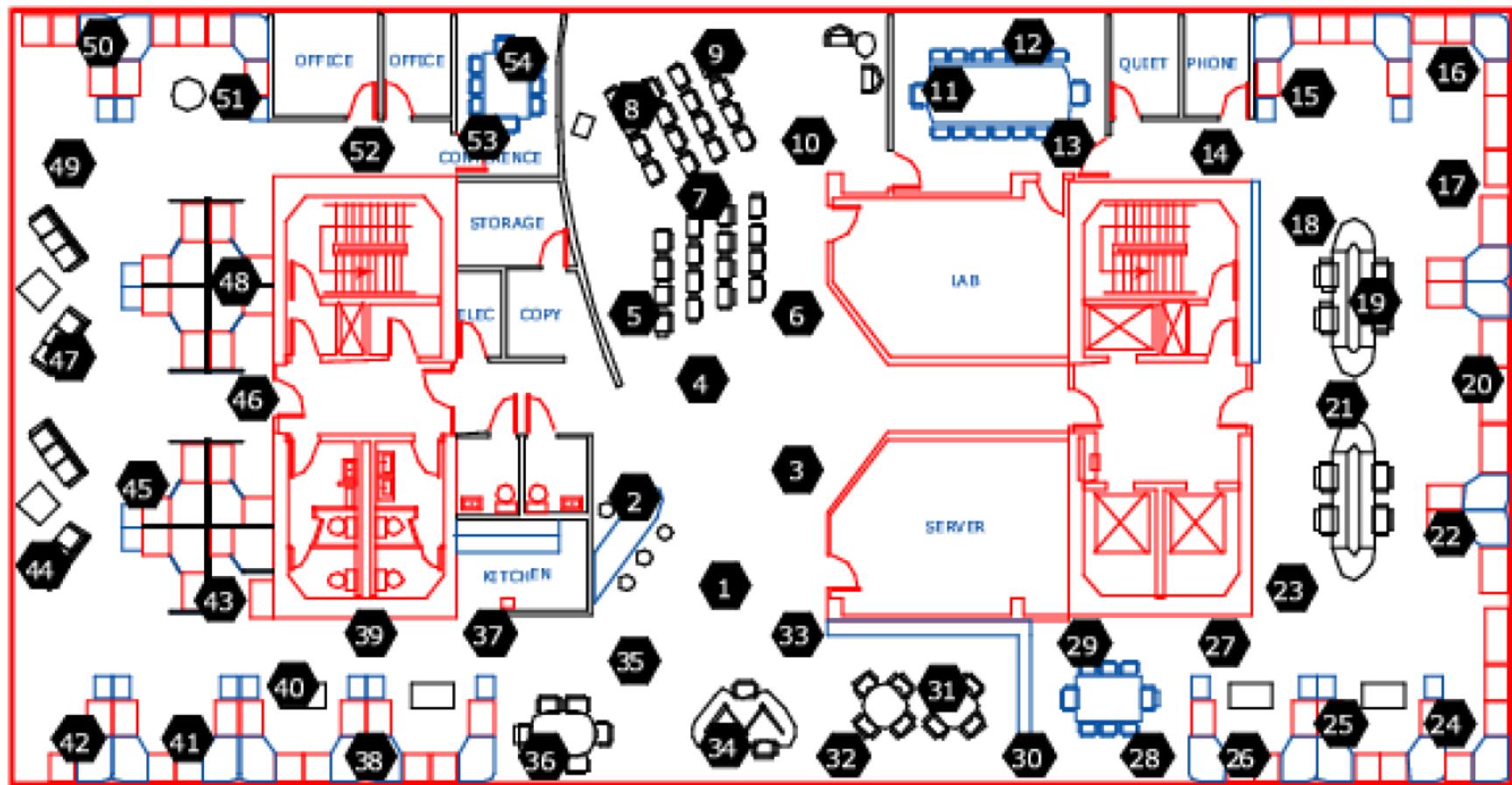
In the PC2 vs PC1 plot, versicolor and virginica are much better separated.

## Revisit Iris



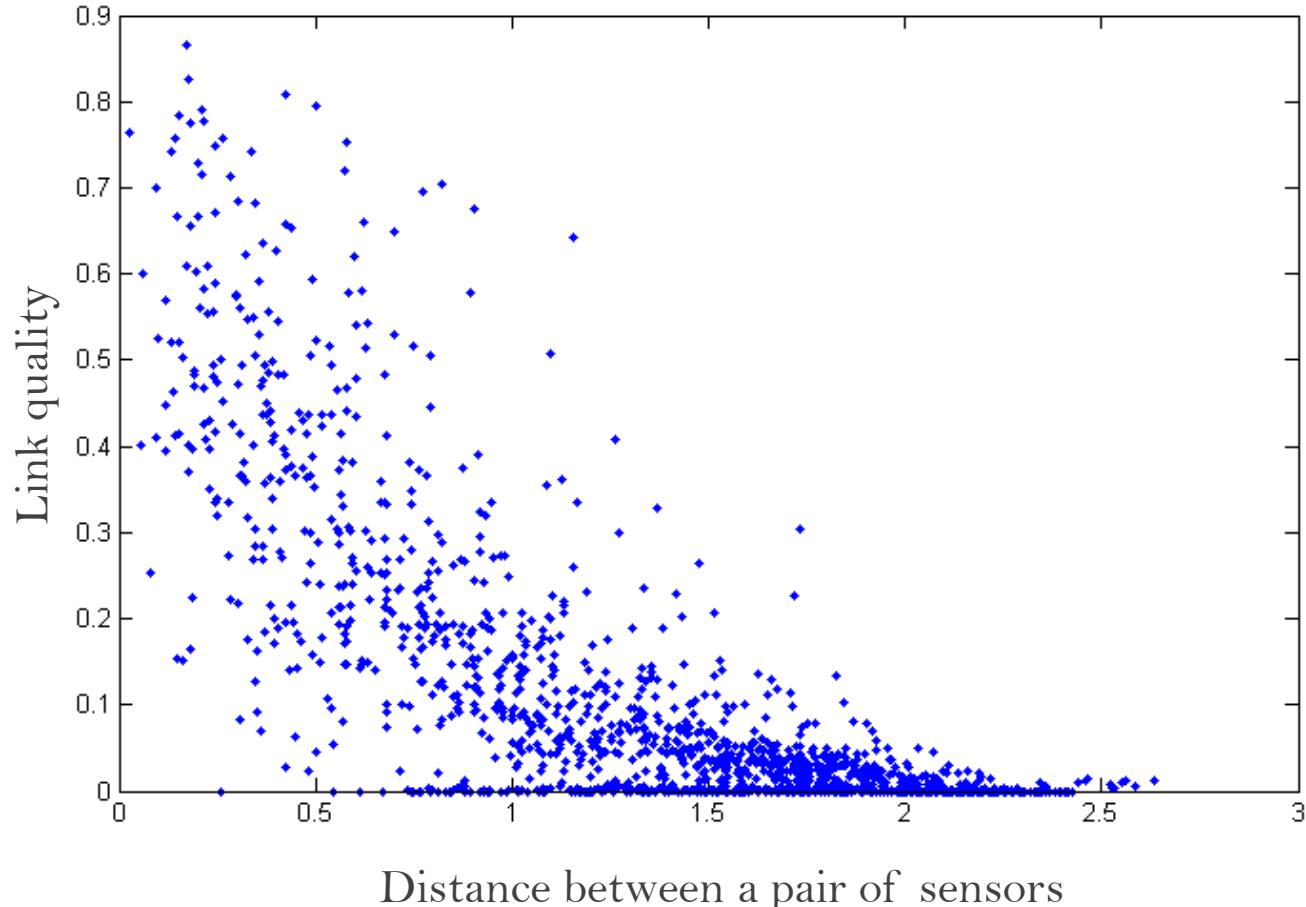
Rotation matrix

# What Does PCA Do? – Magic Time



Sensors in Intel Berkeley Lab

# Pairwise link quality vs. distance



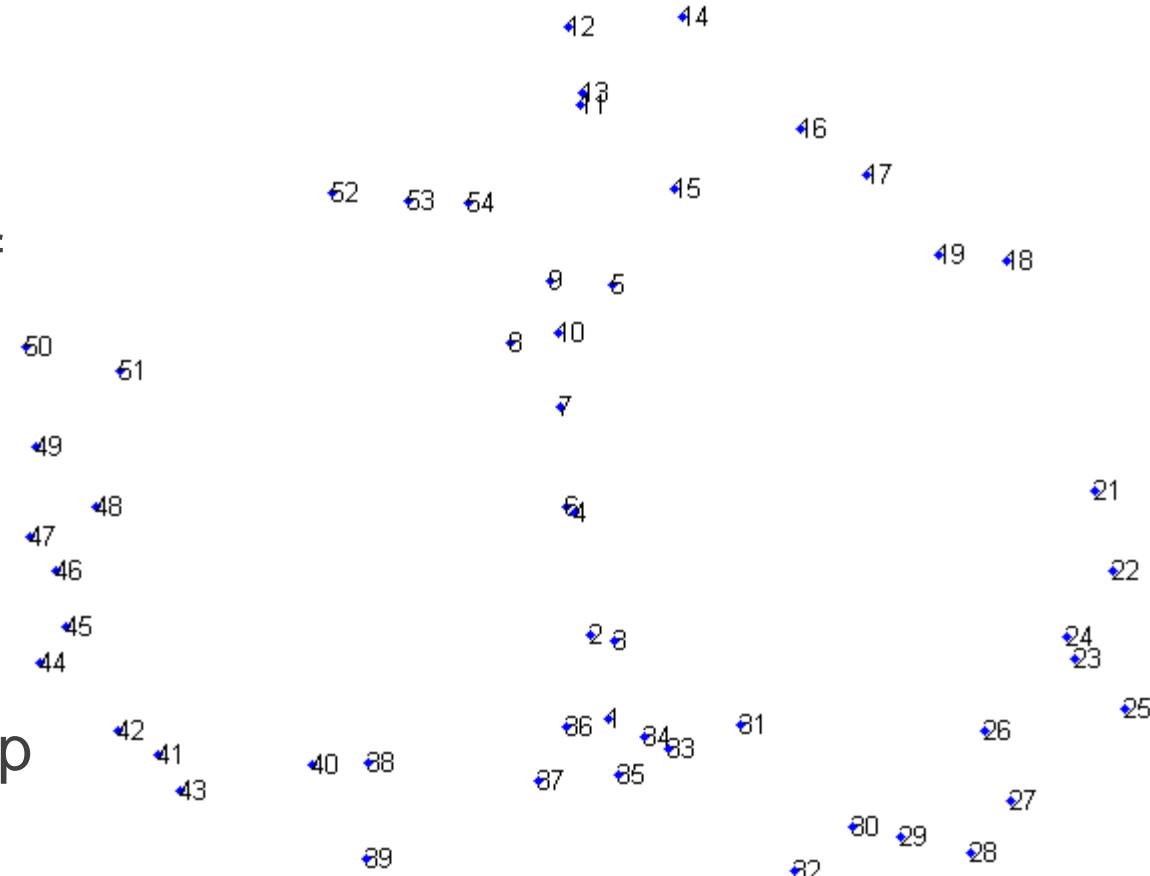
# PCA in action

Given a 54x54 matrix of pairwise link qualities

## Do PCA

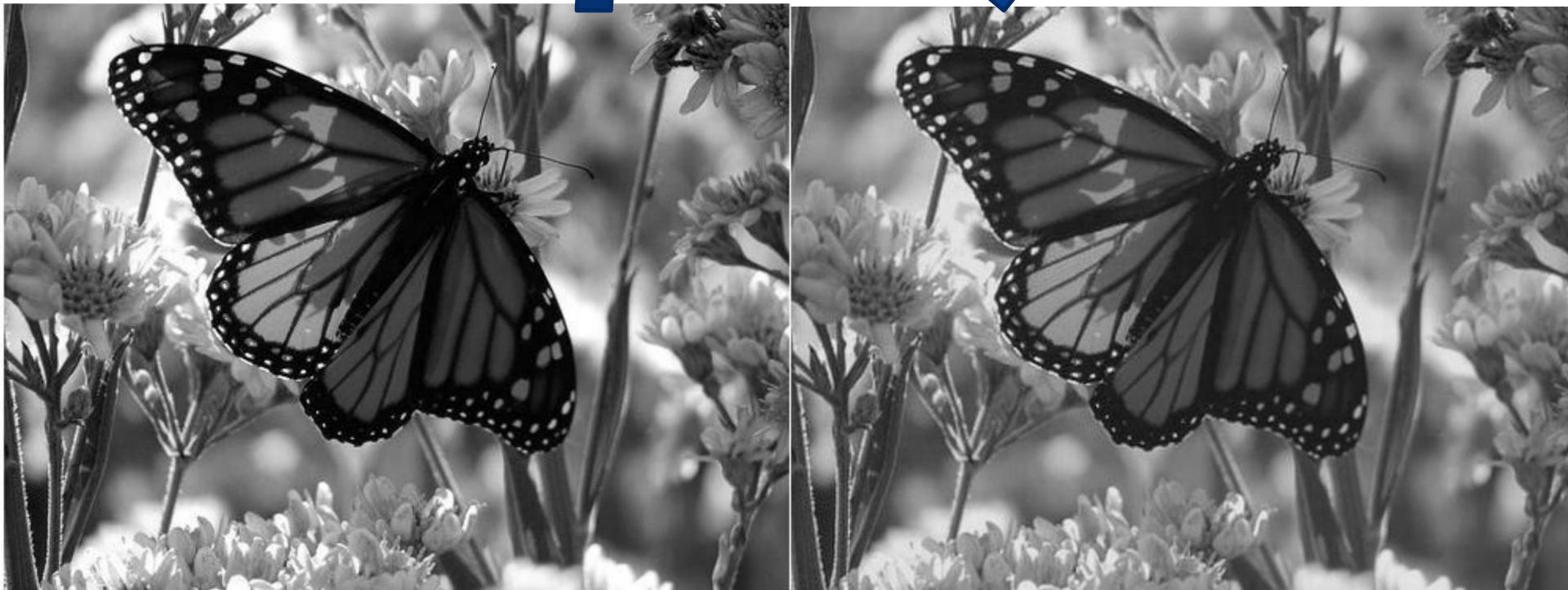
# Project down to 2 principal dimensions

# PCA discovered the map of the lab



# PCA in action

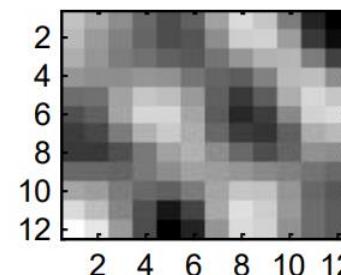
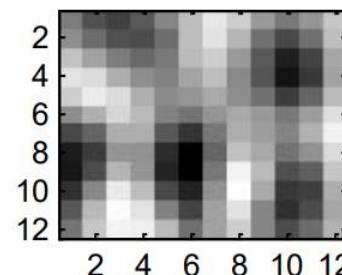
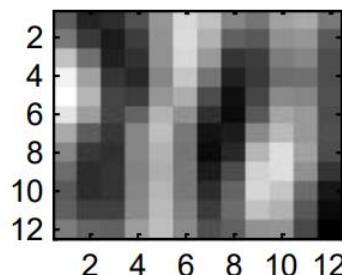
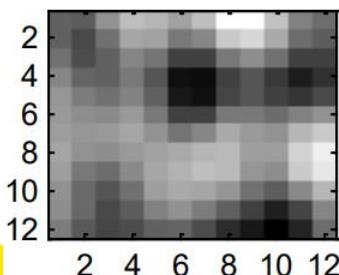
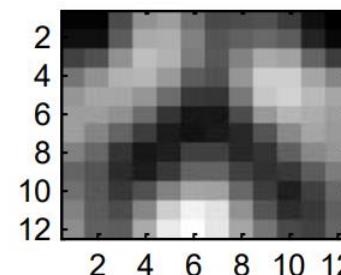
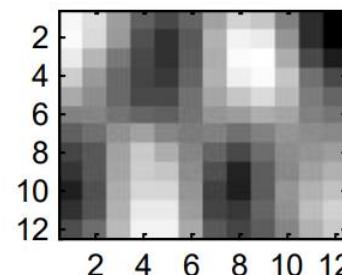
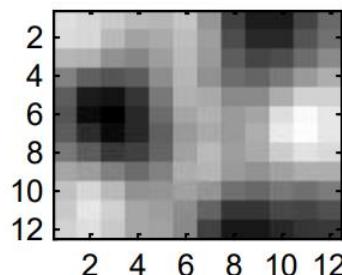
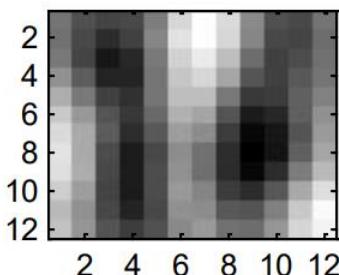
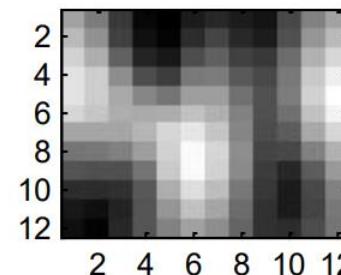
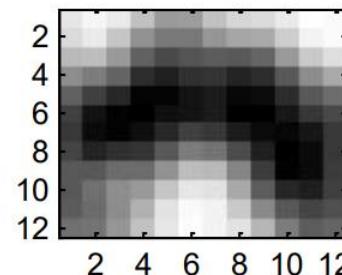
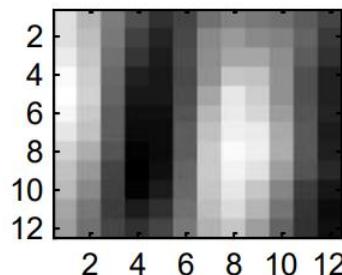
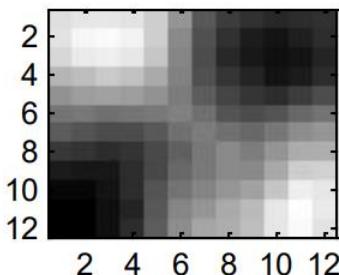
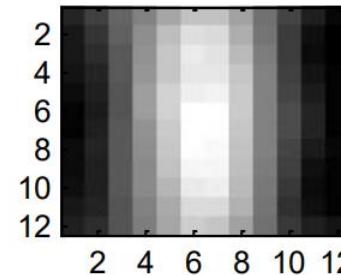
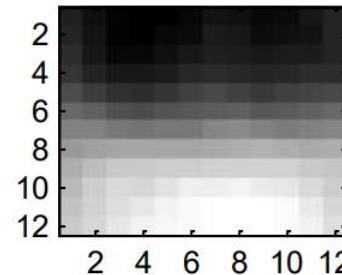
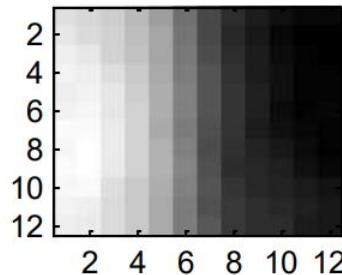
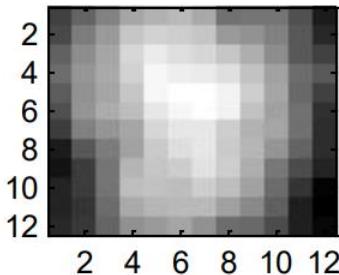
16D



Divide the original 372x492 image into patches:

- Each patch is an instance that contains 12x12 pixels on a grid  
Consider each as a 144D vector

# PCA in action

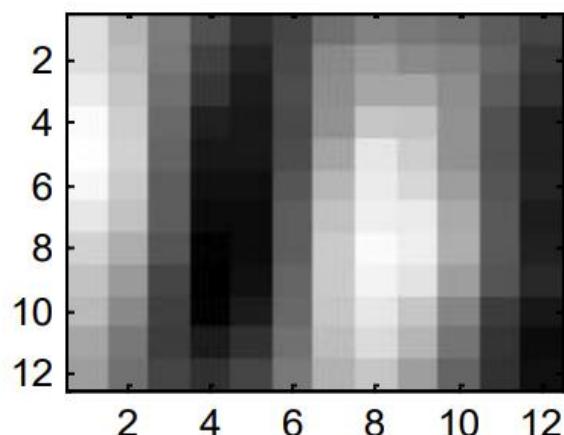
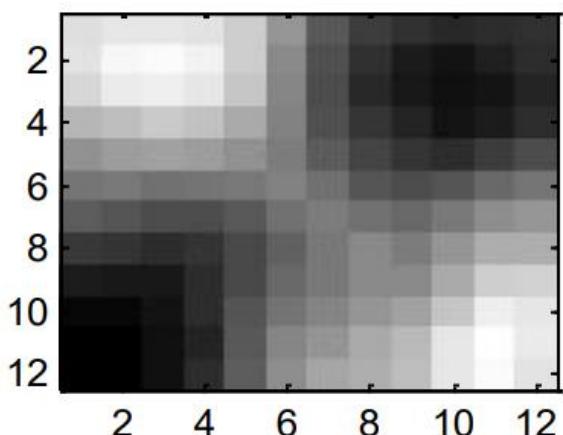
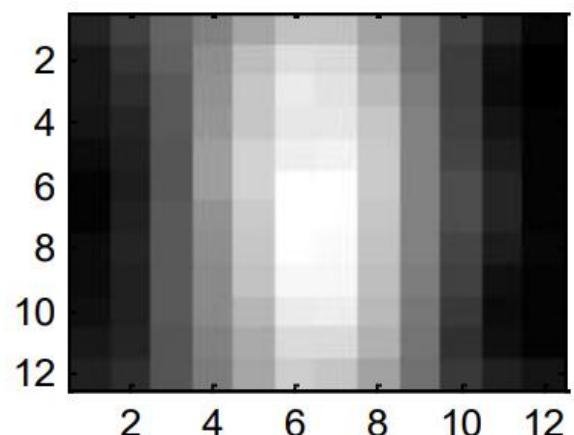
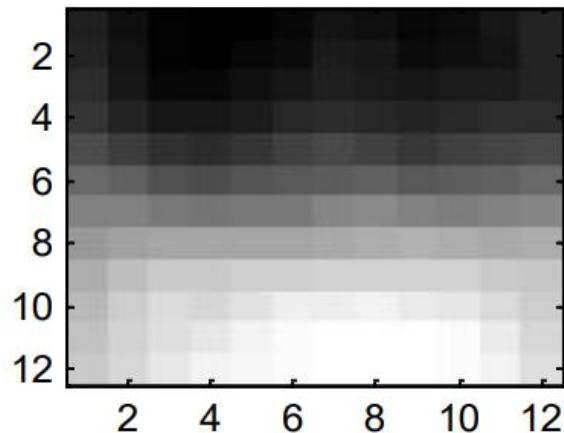
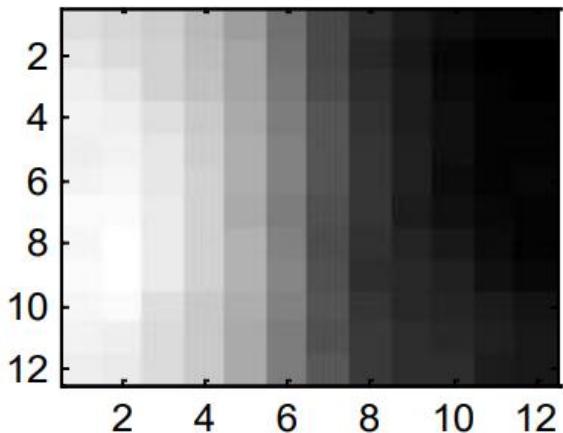
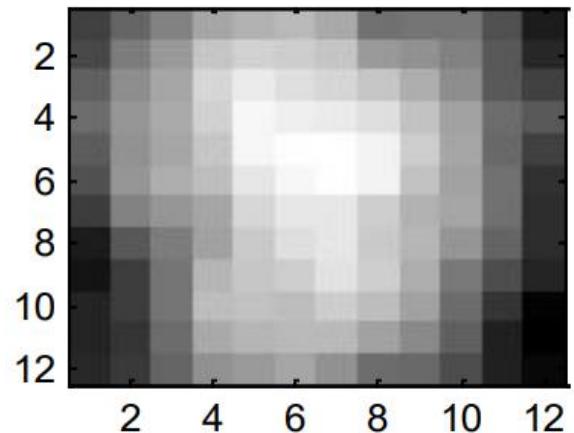


# PCA in action

How about fewer PCs?



# PCA in action

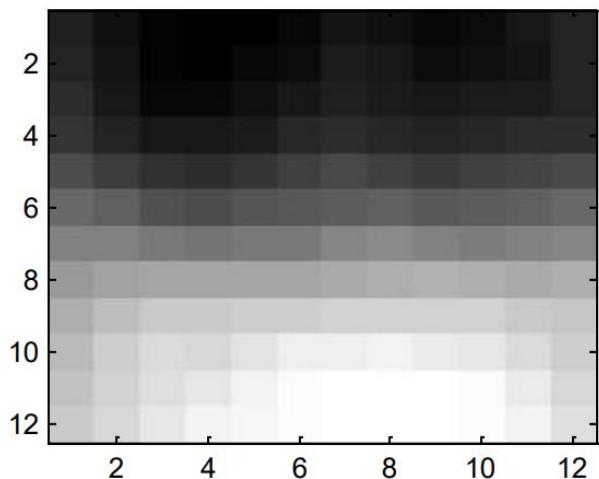
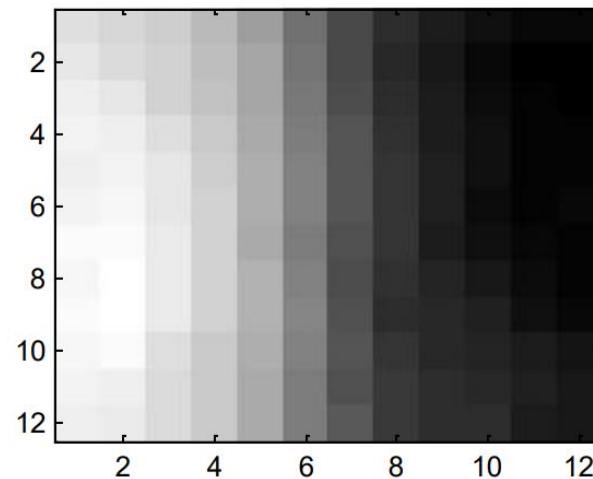
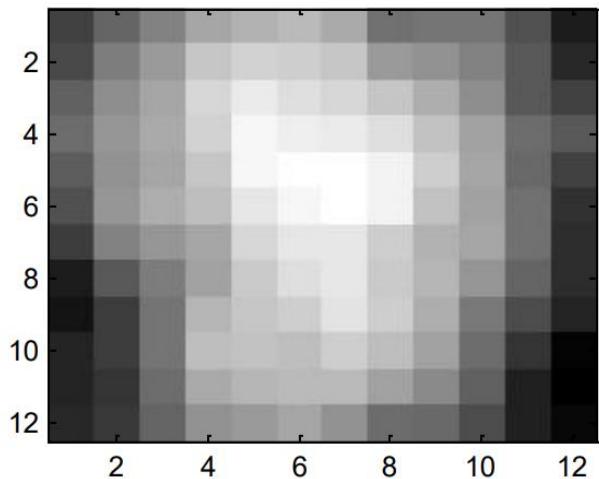


# PCA in action

How about fewer PCs?



# PCA in action



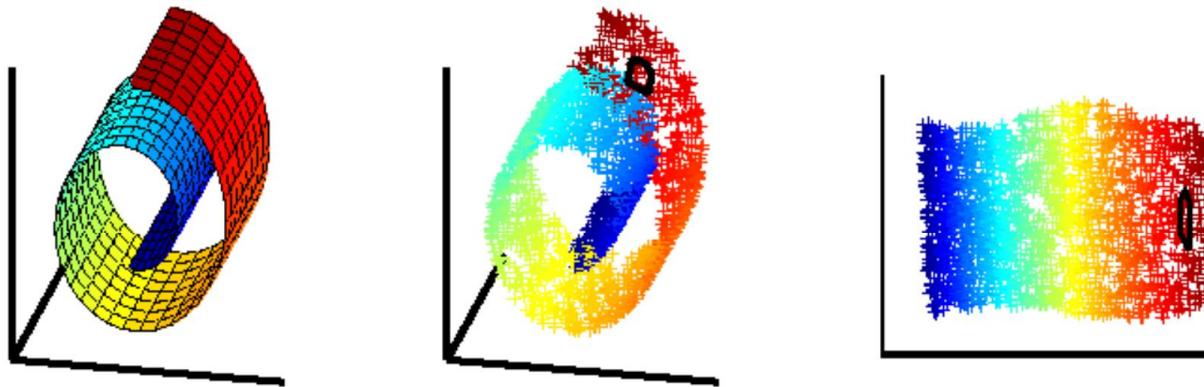
# PCA in action

15 PCs



# More discussions on PCA

1. Can't handle non-flat. Only works on linear variations



2. Sensitive to scaling
3. Insufficiency for some distributions: mean and covariance
4. Assumption might be problematic:

larger variance is equal to more important?