

Managing Big Data

Preliminaries: About Data

Manolis Tzagarakis
Associate Professor
Department of Economics
University of Patras

tzagara@upatras.gr
2610 962588
google:tzagara
Facebook: tzagara
SkypeID: tzagara
QuakeLive: DeusEx
CoD: CoDFather

About data

About data

- What is data? **A collection of objects and their attributes**
- An **attribute** is a **property or characteristic of an object**
 - Examples: eye color of a person, temperature, etc.
 - Attribute also known with different names: variable, field, characteristic, feature or dimension
- A **collection of attributes describe an object**
 - Object is also known as *record, point, case, sample, entity, or instance*

**Objects,
records,
instances,
observations,
tuples**

Attributes/Dimensions

<i>Tid</i>	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

About data

◎ **Attribute values?**

- › Numbers or symbols assigned/mapped to an attribute

◎ **Attribute vs Attribute values**

- › Same attribute can be mapped to different attribute values
 - E.g. height in feet or meters
- › Different attributes can be mapped to the same set of values
 - E.g. Attribute values for ID and age are integers
 - But with different properties: id's don't have limits, age has

About data

- ◉ There are different types of attributes based on the values they receive (which determines also **how you can analyse them in terms of operations**):
 - > **Nominal**
 - Examples: ID numbers, eye color, zip codes
 - > **Ordinal**
 - Examples: rankings (e.g., taste of potato chips on a scale from 1-10 – Likert scale), grades, height in {tall, medium, short}
 - > **Interval**
 - Examples: calendar dates, temperatures in Celsius or Fahrenheit.
 - > **Ratio**
 - Examples: temperature in Kelvin, length, time, counts

About data

- ◎ The **type of an attribute** depends on which of the following properties it possesses (basically **what arithmetic operations** you can do with them):
 - > **Distinctness:** = ≠
 - > **Order:** < >
 - > **Addition:** + -
 - > **Multiplication:** * /

 - > **Nominal attribute:** distinctness
 - > **Ordinal attribute:** distinctness & order
 - > **Interval attribute:** distinctness & order & addition
 - > **Ratio attribute:** distinctness & order & addition & Multiplication

		Attribute Type	Description	Examples	Allowed Operations
Qualitative		Nominal	The values of a nominal attribute are just different names, i.e., nominal attributes provide only enough information to distinguish one object from another. (=, ≠)	zip codes, employee ID numbers, eye color, sex: { <i>male</i> , <i>female</i> }	mode, entropy, contingency correlation, χ^2 test
		Ordinal	The values of an ordinal attribute provide enough information to order objects. (<, >)	hardness of minerals, {good, better, best}, grades, street numbers, Likert scales	median, percentiles, rank correlation, run tests, sign tests
Quantitative		Interval	For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. (+, -) but not (*, /) . E.g. 30°C is not twice as hot as 15°C	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, t and F tests
		Ratio	For ratio variables, both differences and ratios are meaningful. (*, /)	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

Allowed transformation i.e. transformation that do not change the meaning of the attribute

Attribute Level	Allowed Transformation	Comments
Nominal	Any permutation of values	If all employee ID numbers were reassigned, would it make any difference?
Ordinal	An order preserving change of values, i.e., $new_value = f(old_value)$ where f is a monotonic function.	An attribute encompassing the notion of good, better best can be represented equally well by the values {1, 2, 3} or by {0.5, 1, 10}.
Interval	$new_value = a * old_value + b$, where a and b are constants	E.g. the Fahrenheit and Celsius temperature scales differ in terms of where their zero value is and the size of a unit (degree).
Ratio	$new_value = a * old_value$	Length can be measured in meters or feet.

Discrete and continuous

⦿ Discrete Attribute

- > Has only a finite (or countable infinite set) of values (countable means can be ordered with a relationship)
 - > **Examples:** zip codes, counts, or the set of words in a collection of documents, number of birds in a flock
- > Often represented as integer variables.
- > Note: binary attributes are a special case of discrete attributes.

⦿ Continuous Attribute

- > Has real numbers as attribute values (cannot be ordered with a relationship)
 - > **Examples:** temperature, height, weight, salary.
- > Practically, real values can only be measured and represented using a finite number of digits.
- > Continuous attributes are typically represented as floating-point variables.

Types of data sets

- Ways in which they are represented/structured
 - > “Structured” data: ordered/grouped in some particular way which (structure) is understandable to humans AND machines.
- **Record data**
 - > Data Matrix
 - > Document Data
 - > Transaction Data
- **Graph data**
 - > World Wide Web
 - > Molecular Structures
- **Ordered data**
 - > Spatial Data
 - > Temporal Data
 - > Sequential Data
 - > Genetic Sequence Data

Important characteristics of structured data

- ◉ Dimensionality

- > How many dimensions the data has (here **dimensions**: number of variables, features, attributes)
- > Dimensionality is a big problem (curse of dimensionality)

- ◉ Sparsity

- > How many values are present (or non-present/zero)?

- ◉ Resolution

- > Different patterns at different scales

Record data

- Record: a **fixed set of attributed**, handled as one entity
- Record data: collection of records

Record data

<i>Tid</i>	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

One record

Data matrix

- If data objects have the same fixed set of numeric attributes, then the data objects **can be thought of as points in a multi-dimensional space**, where each dimension represents a distinct attribute
- Such data set can be represented by an m by n matrix, where there are m rows, one for each object, and n columns, one for each attribute

Projection of x Load	Projection of y load	Distance	Load	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

Document data

- Each document becomes a **'term' vector** creating collectively a **Document Term Matrix**.
 - each term is a component (attribute) of the vector,
 - the value of each component is the number of times the corresponding term occurs in the document.

Note: Search engines like google, Bing do this

Term/lemma/word **'team'** appears in **'Document 1'** 3 times



	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

Document Term Matrix

Term vector. Vectorizing a document

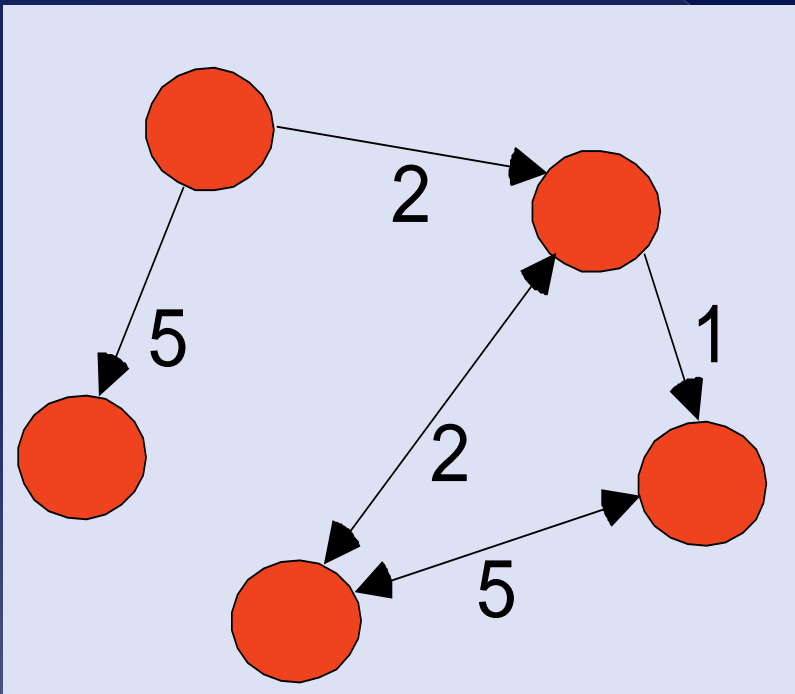
Transaction data

- A special type of record data, where
 - > each record (transaction) involves a set of items.
 - > For example, consider a **grocery store**. The set of products purchased by a customer during one shopping trip constitute a transaction, while the individual products that were purchased are the items.

<i>TID</i>	<i>Items</i>
1	Bread, Coke, Milk
2	Beer, Bread
3	Beer, Coke, Diaper, Milk
4	Beer, Bread, Diaper, Milk
5	Coke, Diaper, Milk

Graph data

- Data represented with nodes and links
(=**graphs**)



- **Examples**

- > World wide web (pages, links)
- > References in scientific articles
 - Who references which paper (link)
- > Calculate
 - **PageRank (google)**
 - **h-index**
 - **Hubs**

Ordered data

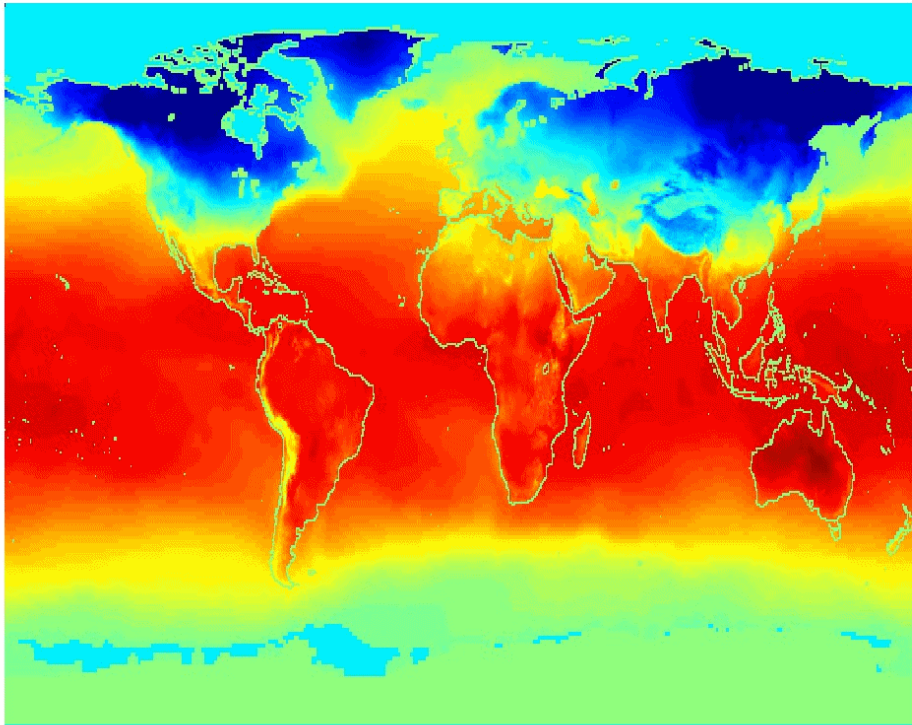
- ◎ Position/rank matters
 - > E.g. genomic sequence

```
GGTTC CGCCTTCAGCCCCGCGCC  
CGCAGGGCCCGCCCCGCGCCGTC  
GAGAAGGGCCCGCCTGGCGGGCG  
GGGGGAGGCGGGGCCGCCCGAGC  
CCAACCGAGTCCGACCAGGTGCC  
CCCTCTGCTCGGCCTAGACCTGA  
GCTCATTAGGCGGCAGCGGACAG  
GCCAAGTAGAACACGCGAAGCGC  
TGGGCTGCCTGCTGCGACCAGGG
```

Ordered data

- Spatio-temporal data

Jan



Average
Monthly
Temperature of
land and ocean

Data quality

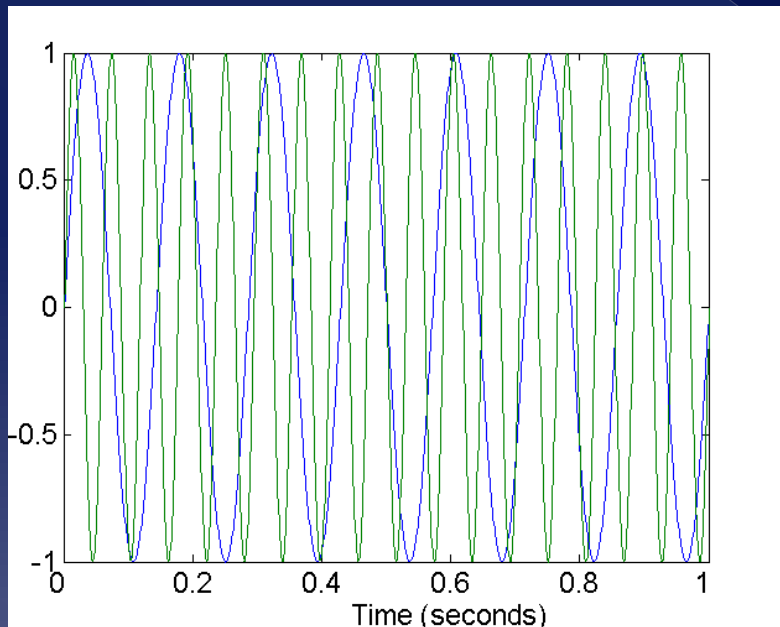
Data quality

◎ Data quality?

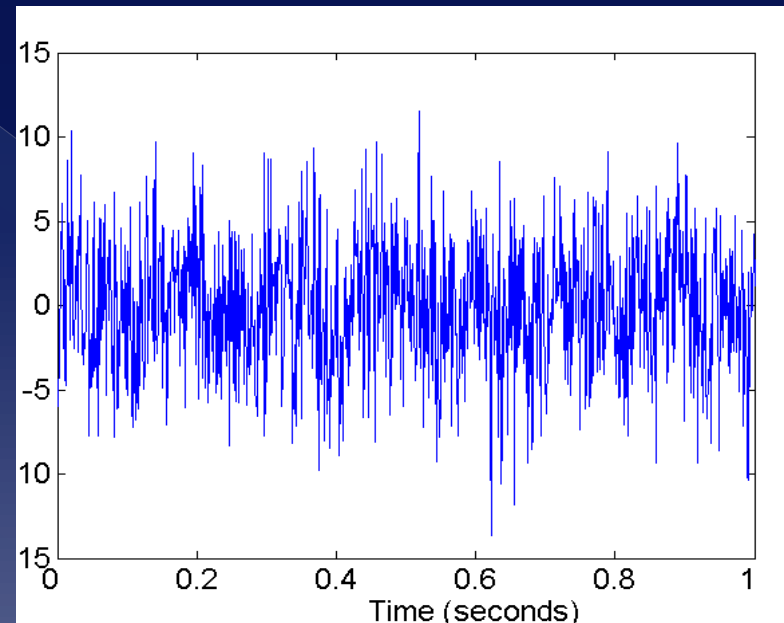
- > The aspects of data sets that make it **suitable/useful** (or not) **for processing to achieve a goal**
- > Common data quality issue/problems
 - Noise and outliers
 - Missing values
 - Duplicate data

Noise

- Noise refers to **involuntarily modification** of **original values**
 - Examples: distortion of a person's voice when talking on a poor phone and "snow" on television screen



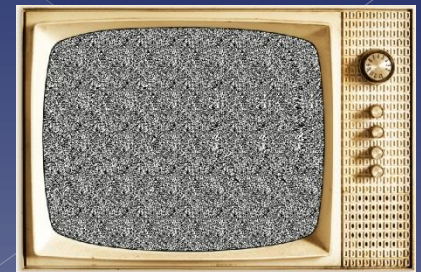
Two Sine Waves (voice)



Two Sine Waves + Noise (voice + distortion). Yup it's a miracle that you can hear a voice.

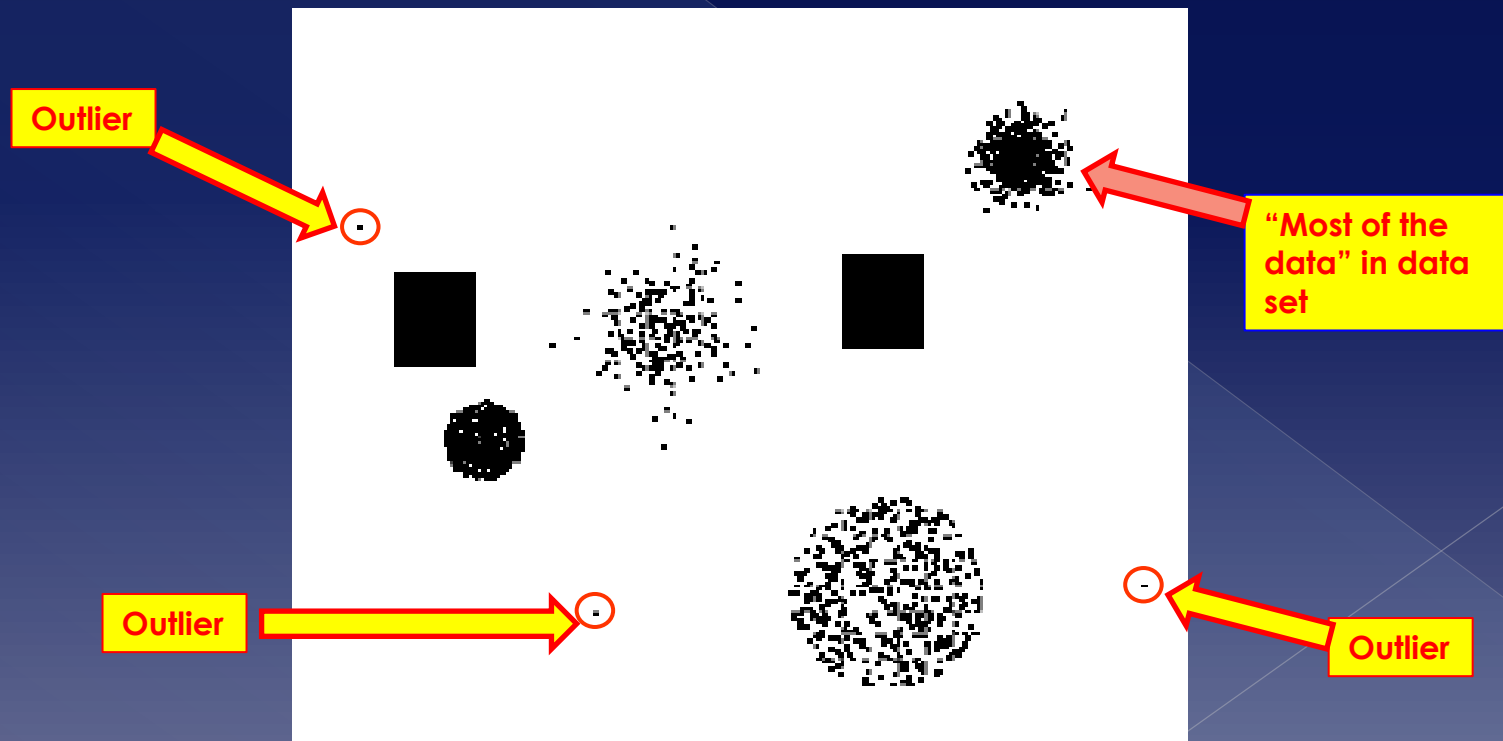
Noise

- ◉ Useless/funny bits of noise
 - > Give rise to **information theory**
 - Claude Shannon aiming at separate voice (information) from not-voice (noise)
 - > Noise **can be interesting**
 - “Snow” on old TVs is **~40% cosmic radiation (from Big Bang)**
 - Meaning the **“snow” that you saw**, was **the cosmos/universe on your TV**
 - Not anymore though due to digital TV.



Outliers

- Outliers are data objects **with characteristics** that **are considerably different** than most of the other data objects in the data set
 - Practical rule: Normally distributed data, everything that's more than 3 standard deviation away from the mean is suspect of an outlier.
 - Special graphs to visualize them e.g. Box plot (Θηκόγραμμα)



Outliers

- Can seriously distort your view when analyzing data even in the simplest way
 - > E.g. finding mean/average

Person	Income (in dollars)
Bill	100000000000
Jim	19
John	20
Phil	25
Martha	16

(Naïve) average income:
200000016 dollars

*#LOL, one person made everyone rich.
No joke pal.*

Missing values

- Reasons for missing values
 - > Information is **not collected** (e.g., people decline to give their age and weight)
 - > Attributes may **not be applicable** to all cases (e.g., annual income is not applicable to children)
 - > Devices may be **faulty** (e.g. faulty thermometer)
- Handling missing values
 - > **Eliminate** missing values in some way (commonly used – many techniques)
 - > **Ignore** the Missing Value During Analysis
 - > **Replace** with specific values (e.g. average) and ways (e.g. weighted by probabilities)

Missing values

- ◉ In real datasets, missing values **represented in various ways**
 - > No value at all (empty value), specific individual values such as ? - _ etc
 - > Empty values are **verified/discovered** by examining the data prior to any analysis
 - During preprocessing or exploratory data analysis stage
 - > May also **read the documentation** to see how missing values are represented in datasets

Missing values

◉ Examples

- > Python and R offer special values to denote that values are missing
 - R: Values **NA** (not available) and **NaN** (Not A Number)
 - Python: **None** and **NaN** (Not A Number)
- > R and Python **recognize** such values **automatically** as meaning missing values and can reason with these

Missing values

csv file

```
Lecture2-taxpayersData.csv — Lectures
1 name,address,salary,ownscar,ownhouse,evades taxes
2 Jim,Mulholland drive,24023.78,y,n,n
3 Maria,Baker Str 221B,26933.01,y,y,n
4 Nik,Downing Street 10,,y,y,y
5 Alice,Wallaby Way 42,13765.45,n,n,n
6 Sonia,742 Evergreen Terrace,22098.04,y,y,y
7 Mark,1600 Pennsylvania Ave,19307.87,n,y,y
```

Empty value
indicating
missing value

Reading above csv file in Python into a data frame:

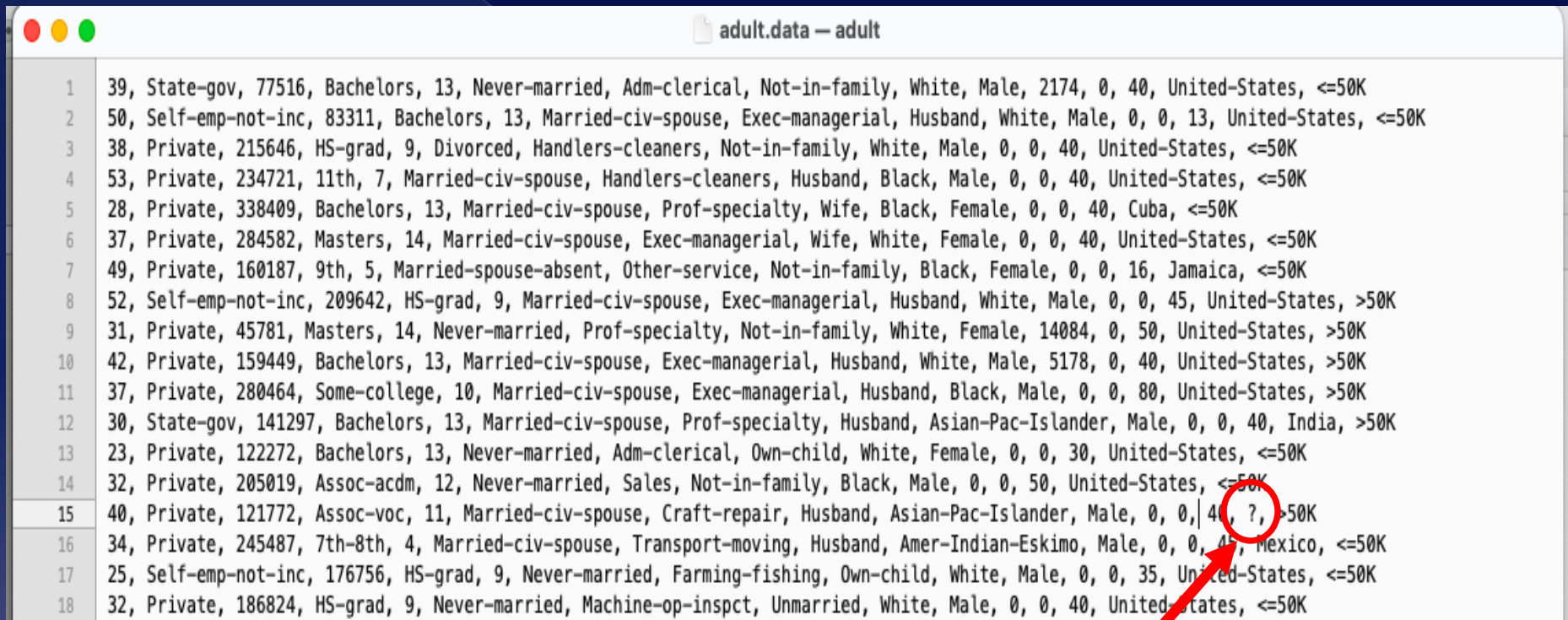
```
import pandas as pd
taxpayerData = pd.read_csv('Lecture2-taxpayersData.csv', header=0, sep=',')
```

```
>>> data
   name      address  salary ownscar ownhouse evades taxes
0  Jim  Mulholland drive  24023.78      y        n          n
1  Maria    Baker Str 221B  26933.01      y        y          n
2  Nik    Downing Street 10      NaN      y        y          y
3  Alice    Wallaby Way 42  13765.45      n        n          n
4  Sonia  742 Evergreen Terrace  22098.04      y        y          y
5  Mark  1600 Pennsylvania Ave  19307.87      n        y          y
>>> |
```

Representation of empty value in
Python. Denotes a missing value

Missing values

- Can be represented in csv files using special values



Line	CSV Row
1	39, State-gov, 77516, Bachelors, 13, Never-married, Adm-clerical, Not-in-family, White, Male, 2174, 0, 40, United-States, <=50K
2	50, Self-emp-not-inc, 83311, Bachelors, 13, Married-civ-spouse, Exec-managerial, Husband, White, Male, 0, 0, 13, United-States, <=50K
3	38, Private, 215646, HS-grad, 9, Divorced, Handlers-cleaners, Not-in-family, White, Male, 0, 0, 40, United-States, <=50K
4	53, Private, 234721, 11th, 7, Married-civ-spouse, Handlers-cleaners, Husband, Black, Male, 0, 0, 40, United-States, <=50K
5	28, Private, 338409, Bachelors, 13, Married-civ-spouse, Prof-specialty, Wife, Black, Female, 0, 0, 40, Cuba, <=50K
6	37, Private, 284582, Masters, 14, Married-civ-spouse, Exec-managerial, Wife, White, Female, 0, 0, 40, United-States, <=50K
7	49, Private, 160187, 9th, 5, Married-spouse-absent, Other-service, Not-in-family, Black, Female, 0, 0, 16, Jamaica, <=50K
8	52, Self-emp-not-inc, 209642, HS-grad, 9, Married-civ-spouse, Exec-managerial, Husband, White, Male, 0, 0, 45, United-States, >50K
9	31, Private, 45781, Masters, 14, Never-married, Prof-specialty, Not-in-family, White, Female, 14084, 0, 50, United-States, >50K
10	42, Private, 159449, Bachelors, 13, Married-civ-spouse, Exec-managerial, Husband, White, Male, 5178, 0, 40, United-States, >50K
11	37, Private, 280464, Some-college, 10, Married-civ-spouse, Exec-managerial, Husband, Black, Male, 0, 0, 80, United-States, >50K
12	30, State-gov, 141297, Bachelors, 13, Married-civ-spouse, Prof-specialty, Husband, Asian-Pac-Islander, Male, 0, 0, 40, India, >50K
13	23, Private, 122272, Bachelors, 13, Never-married, Adm-clerical, Own-child, White, Female, 0, 0, 30, United-States, <=50K
14	32, Private, 205019, Assoc-acdm, 12, Never-married, Sales, Not-in-family, Black, Male, 0, 0, 50, United-States, <=50K
15	40, Private, 121772, Assoc-voc, 11, Married-civ-spouse, Craft-repair, Husband, Asian-Pac-Islander, Male, 0, 0, 40, ?, >50K
16	34, Private, 245487, 7th-8th, 4, Married-civ-spouse, Transport-moving, Husband, Amer-Indian-Eskimo, Male, 0, 0, 45, Mexico, <=50K
17	25, Self-emp-not-inc, 176756, HS-grad, 9, Never-married, Farming-fishing, Own-child, White, Male, 0, 0, 35, United-States, <=50K
18	32, Private, 186824, HS-grad, 9, Never-married, Machine-op-inspct, Unmarried, White, Male, 0, 0, 40, United-States, <=50K

Missing value represented as ? in this dataset (a csv file). Not as an empty value.

Duplicate data

- Data set may include data objects that are duplicates, or **almost duplicates** of one another
 - > Major issue when **merging data from heterogeneous sources (typical in greek public sector)**
- Examples:
 - > Same person with **multiple/different email addresses**
 - Yes, I know what you did on facebook, twitter, insta etc.
- The term is “Data cleaning”
 - > Process of **dealing with duplicate data issues**
 - E.g. names: in greek Κων/νος, Κωνσταντινος, Κώστας, Αγ. Βαρβάρα, Αγία Βαρβάρα etc

Data preprocessing

Data preprocessing

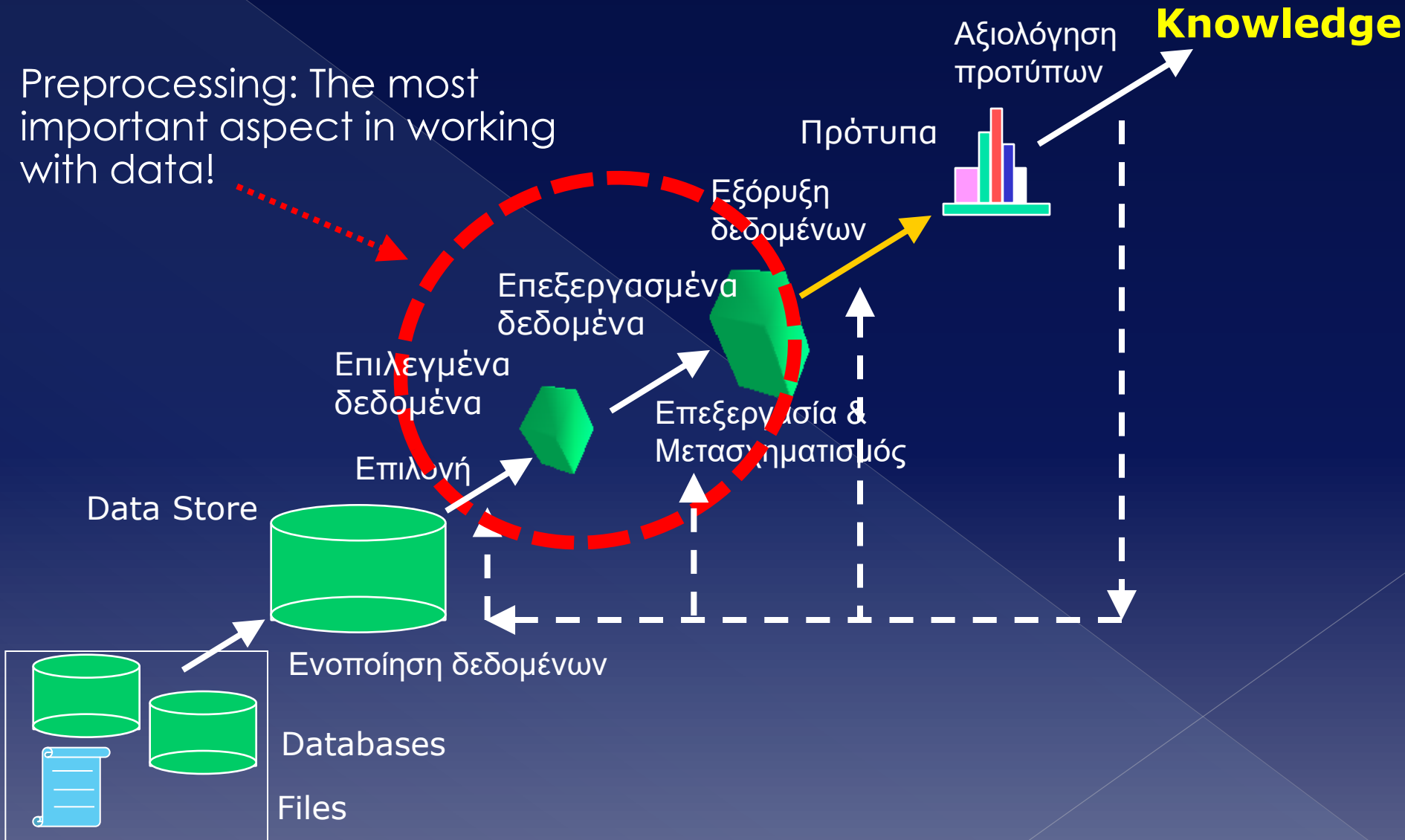
- Data preprocessing?
 - > Steps that aim **making the data**, before their processing, **suitable for the desired processing**.
 - The issue here is to optimize various aspects that may affect processing, in particular
 - **Required space**
 - **Processing time, minimize running times i.e. $O(g(n))$**
 - Don't forget: we're **working with Big data!**
- Probably the most important step in data mining
- In general, **about 70%-80%** of total time is **consumed on data preprocessing tasks**

Data preprocessing

- ⦿ Can definitely shoot your own foot
 - > **Wrong preprocessing yields ALWAYS to wrong results when doing analys. Garbage in-garbage out.**

Preprocessing

Preprocessing: The most important aspect in working with data!



Data preprocessing

- ⦿ Preprocessing techniques/methods
 - > Aggregation
 - > **Sampling**
 - > **Dimensionality reduction**
 - > Feature subset selection
 - > Feature creation
 - > **Discretization**
 - > Attribute transformation

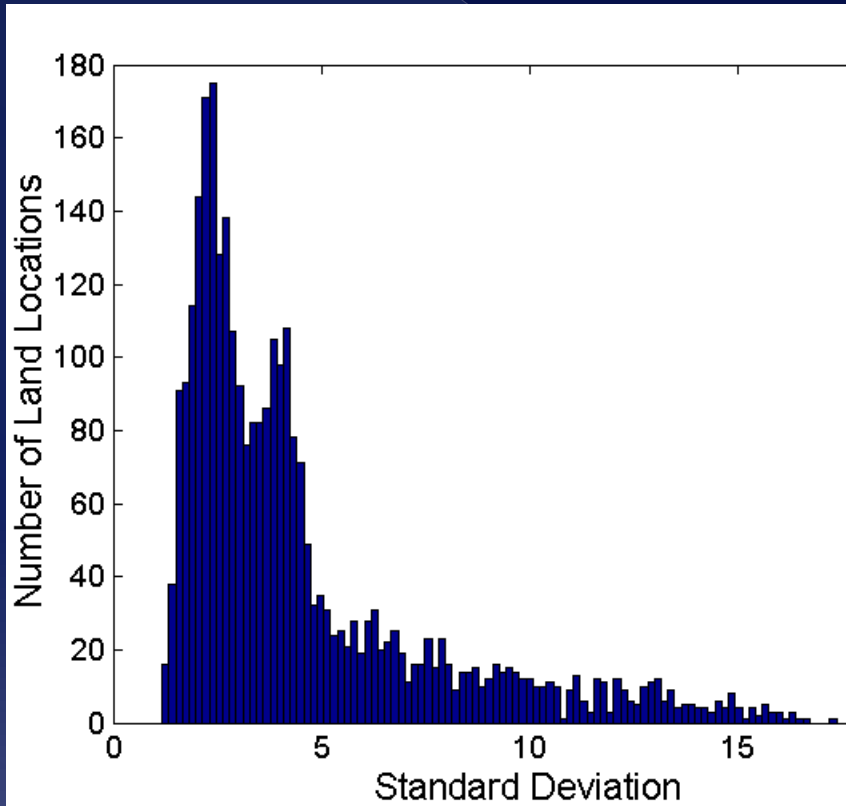
Aggregation

- ◉ Combining two or more attributes (or objects) into a single attribute (or object)

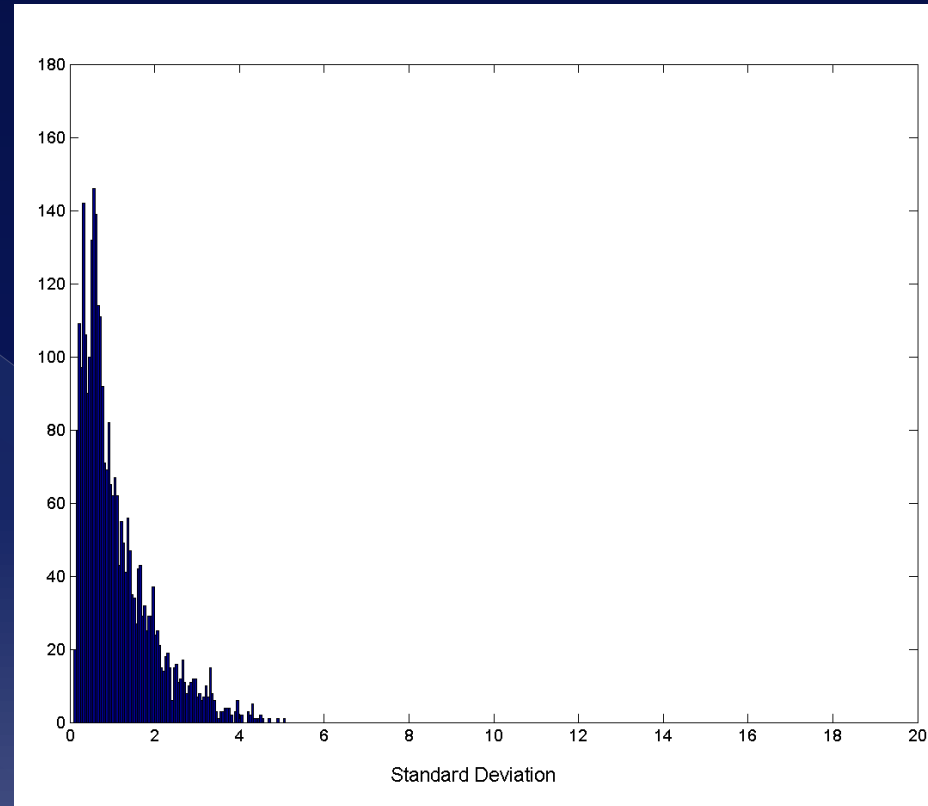
- ◉ Purpose
 - > **Data reduction**
 - Reduce the number of attributes or objects
 - > **Change of scale**
 - Cities aggregated into regions, states, countries, etc
 - > **More “stable” data**
 - Aggregated data tends to have less variability

Aggregation

Variation of Precipitation in Australia



Standard Deviation of
Average Monthly
Precipitation



Standard Deviation of
Average Yearly
Precipitation – **aggregated**
(note smaller variability)

Sampling

- Sampling is the main technique employed for data selection.
 - > It is often used for both the preliminary investigation of the data and the final data analysis.
- Why sampling?
 - > Statisticians sample because obtaining the **entire set** of data of interest is **too expensive or time consuming**.
- Sampling is used in data mining because **processing the entire set** of data of interest is **too expensive or time consuming**.

Sampling

- The key principle for effective sampling is the following:
 - > using a sample will **work almost as well as using the entire data sets, if the sample is representative**
 - > A sample is **representative if it has approximately the same property** (of interest) as the original set of data
 - In terms of its distribution

Sampling

- ◎ **Simple Random Sampling**

- > There is an equal probability of selecting any particular item

- ◎ **Sampling without replacement**

- > As each item is selected, it is removed from the population

- ◎ **Sampling with replacement**

- > Objects are not removed from the population as they are selected for the sample.
 - In sampling with replacement, the same object can be picked up more than once

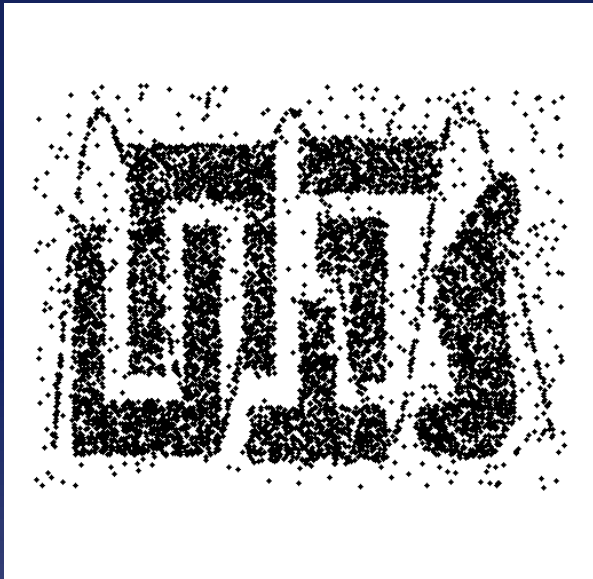
- ◎ **Stratified sampling**

- > Split the data into several partitions; then draw random samples from each partition

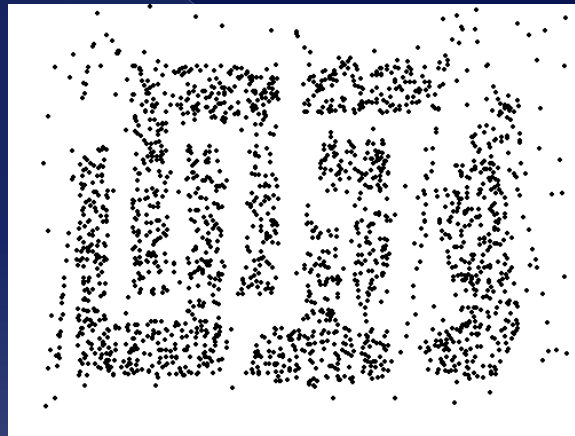
Sampling

- Effect of sample size?

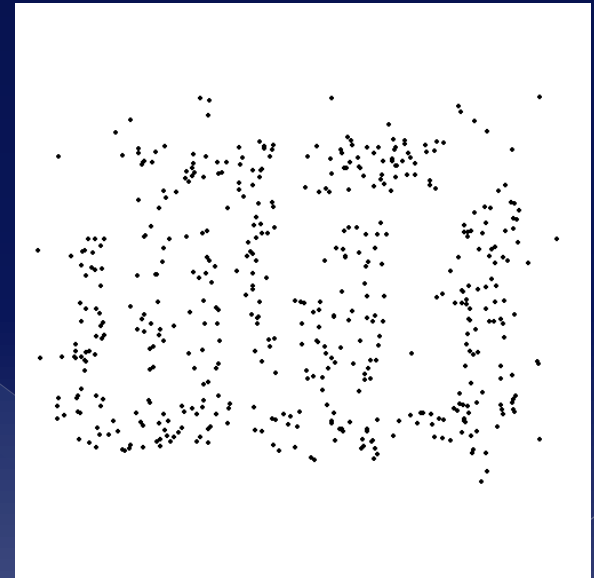
Effects almost any type of data: e.g. when you talk on the phone and your voice is sent to other guy/girl



8000 points



2000 Points

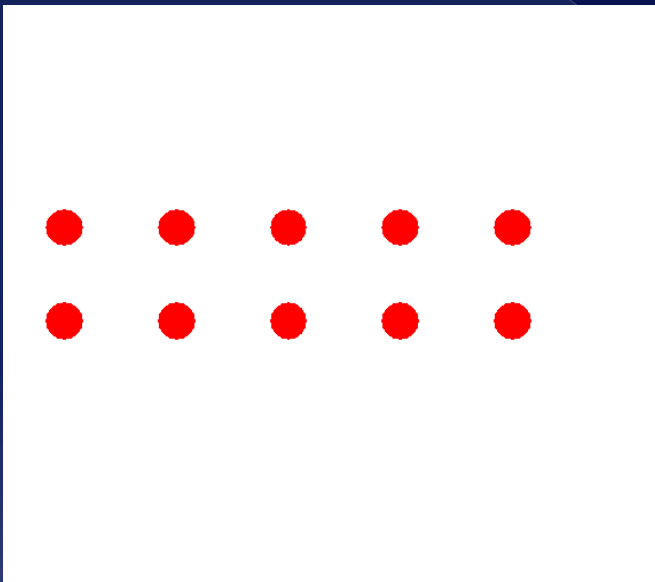


500 Points

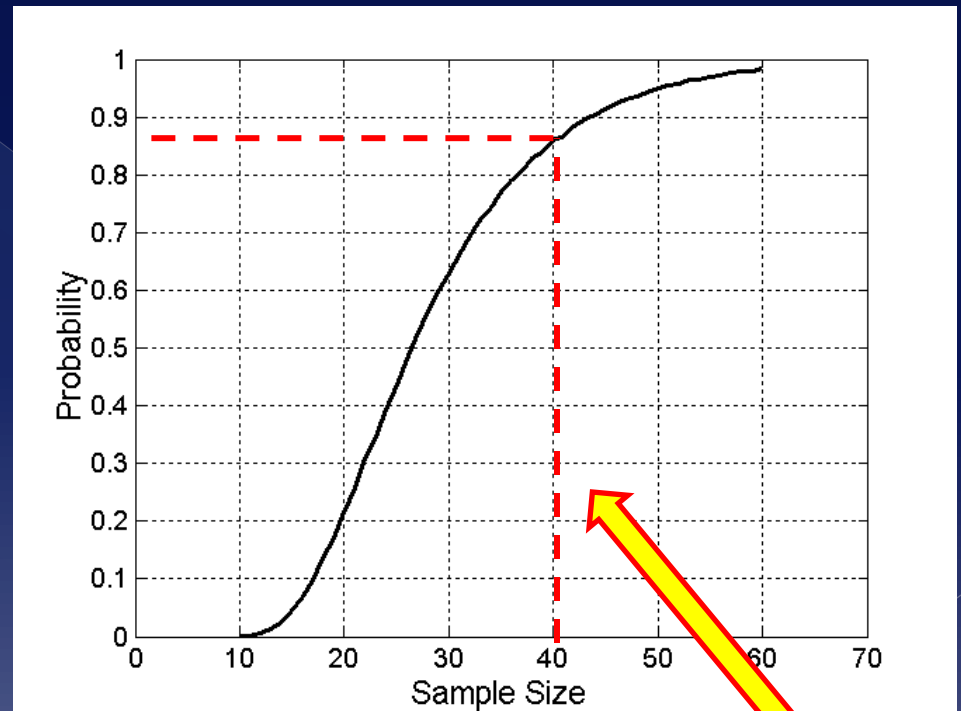
More objects/points/data is in general better. But more objects require more space, more time (preprocessing and analysis). Tradeoff.

Sampling

- What sample size is necessary to get at least one object from each of 10 groups.



10 groups



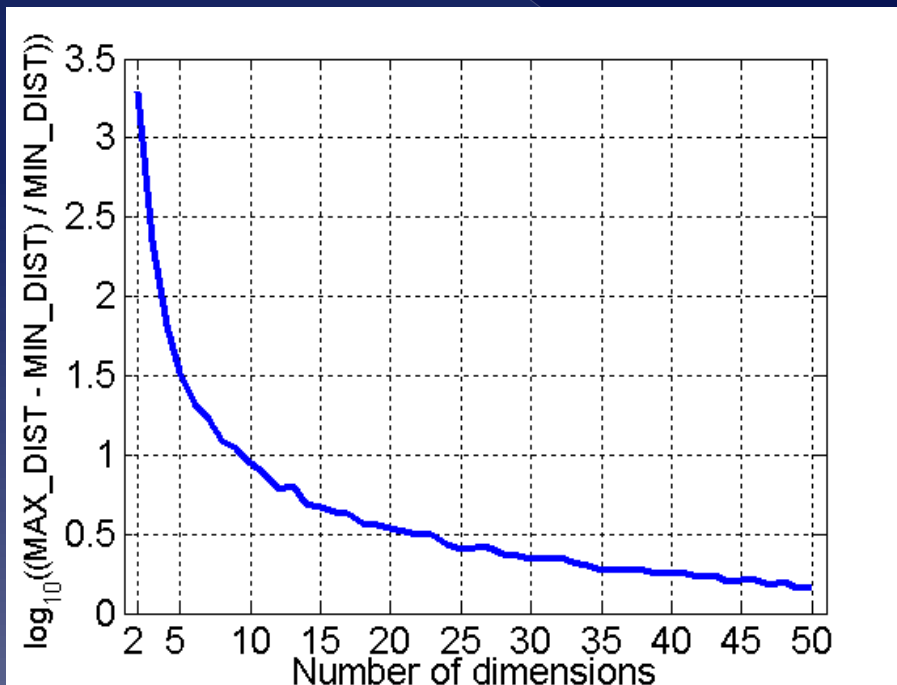
This graph will tell you. For example with a sample size of 40, the probability of having at least one from each group is ~ 0.87 .

Dimensionality reduction

◉ Curse of dimensionality

- > Remember: **dimensions = number of attributes**

- Randomly generate 500 points
- Compute difference between max and min distance between any pair of points
- Note: as dimensions increase, less meaningful distance **which causes problems when clustering!**

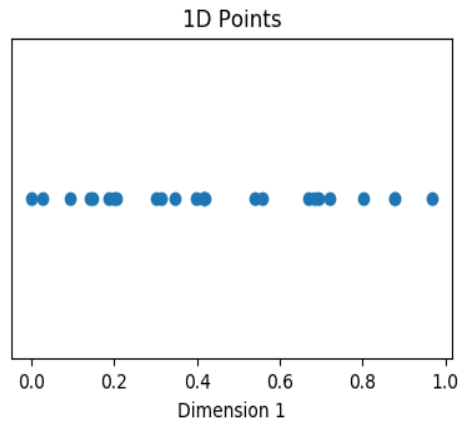


When **dimensionality increases**, data becomes increasingly sparse in the space that it occupies (i.e. many, many, missing or zero values)

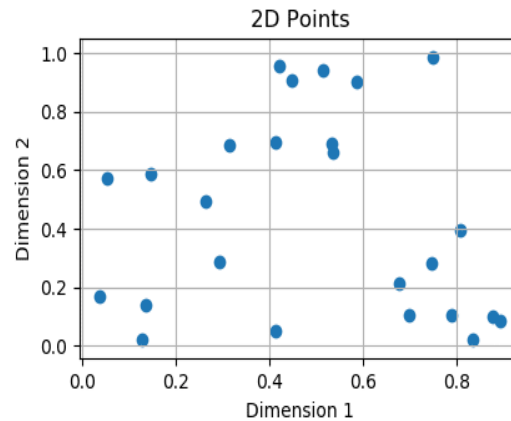
Definitions of **density and distance** between points, which is critical for clustering and outlier detection, **become less meaningful**

Dimensionality reduction

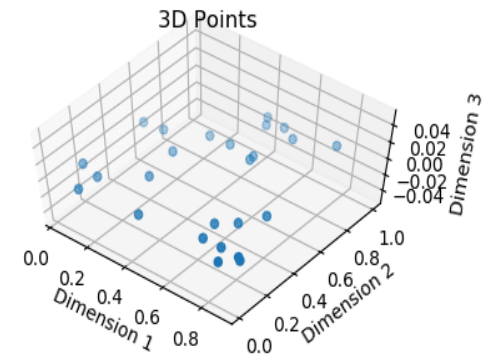
Curse of dimensionality –visual example



1 variable



2 variables



3 variables

- **Same data (25 observations)**, when one more variable (dimension) is added. Look at how the space increases exponentially. Data becomes increasingly sparse inside the space that it occupies (i.e. many, many, missing, zero or absent values). **This means that when dimensions increase but not your number of observations, statistical tests lose power. Many (much) more observations are required.**

Dimensionality reduction

- ⦿ **Due to the curse of dimensionality**, you need at least a **minimum number of observations in your dataset**, in order to get reliable results from any statistical test or any machine learning algorithm.
 - > I guess you've heard that.
 - > **Rule of thumb:** in machine learning, at least 5 different values for each variable.

Dimensionality reduction

- ◎ Purpose of dimensionality reduction
 - > Overcome dimensionality curse (ha, take that!)
 - > **Reduce space and time required** by data processing algorithms
 - Data too Big for your machine
 - > **Facilitate easy visualization** of data – i.e. drawing graphs as a first look at your data
 - > May help in reducing noise
- ◎ Techniques/methods
 - > **Principal Components Analysis (PCA)**
 - > Singular Value Decomposition (SVD)
 - > Supervised and non-supervised techniques

Data preprocessing

PCA – Principal Component Analysis

Principal Components Analysis (PCA)

- How to reduce the dimension of a dataset, if it's very large?
 - > E.g. hundreds/thousands of attributes/variables?
 - > You just **can't drop randomly some** of the variables
 - What if you drop/leave out the important ones?
 - Which one are important, which don't?
 - In addition, you want also **to minimize the number of variables** but at the same time keep all the important statistics of your dataset e.g. variance.

Principal Components Analysis (PCA)

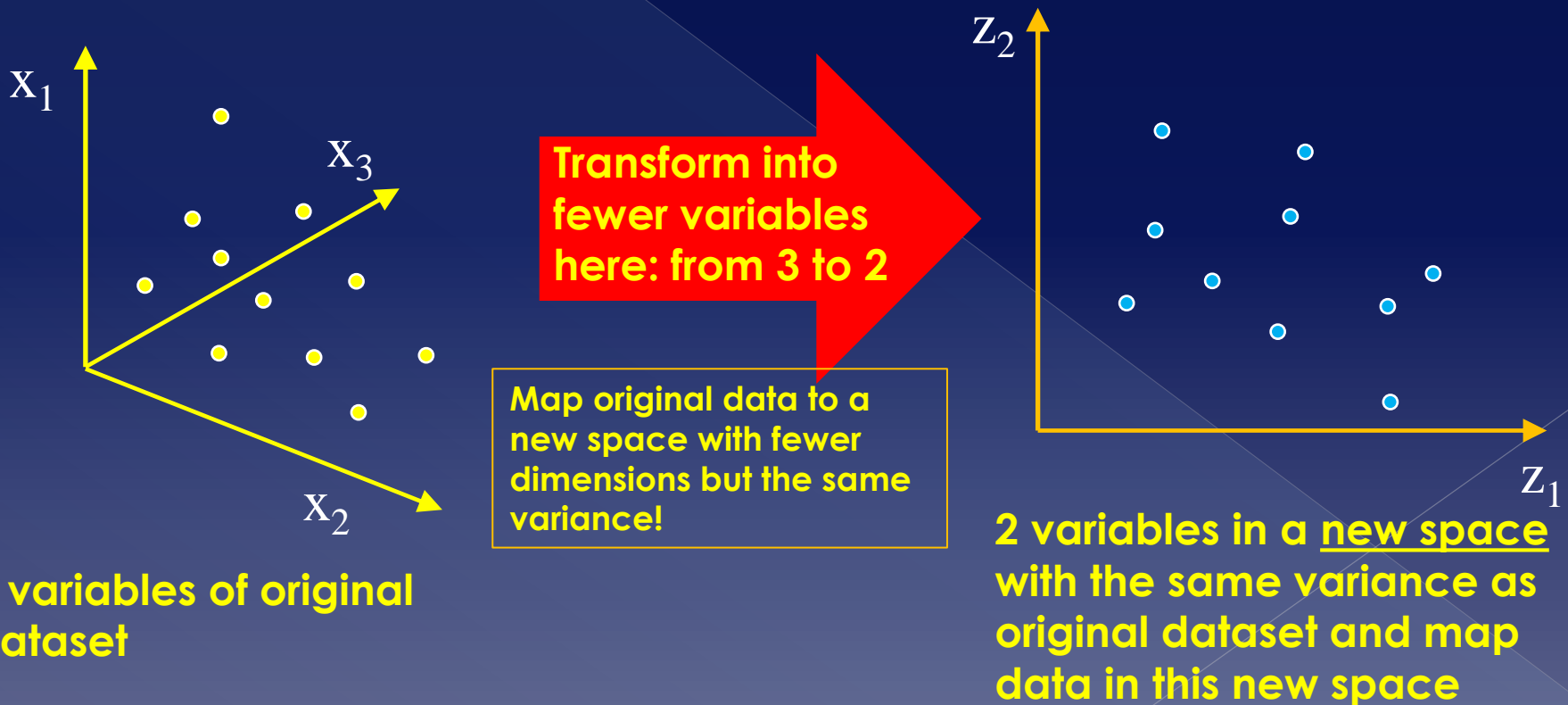
- One approach of doing this is to consider your **original data with the many variables** as points in a space where each variable is a dimension in that space and **transform** this data into data of another space **with fewer dimensions (variables)** but the **same statistical properties**.
 - > And in particular try to **keep** in that new space each variable's variance in the original dataset!

Principal Components Analysis (PCA)

- ◉ But **why keep (or explain) variance** of variables in that new space and **not e.g. mean or any other statistic?**
 - > **Variance is a very, very important aspect** of your data. It's the juicy part.
 - > **Variance tells you how your data varies (goes up and down)** – data with no variance is not interesting – and hence is **more interesting in investigating relationships between variables.**
 - > A lot of existing methods target variance
 - **Linear regression** – explain variance
 - **ANOVA** – compare means by analyzing variance
 - And many, many other...

Principal Components Analysis (PCA)

- So, what we are trying to do is to reduce the variables like the following (note we show here 3 original variables to be reduced to 2 – imagine having 100 variables reducing them to 5 or 10 or 20):



Principal Components Analysis (PCA)

- How to find the new axes (which are in essence variables) of this space which explains most of the variance of the original dataset?
 - > This is what **Principal Component Analysis - PCA** does!

Principal Components Analysis (PCA)

- A method to do this is PCA
 - > What it aims for?
 - It aims to **expressing existing data with high dimensionality (variables/attributes, n)** in the context of a new (optimal) axis system (“subspace”) with fewer dimensions d , i.e. $d < n$.
 - **Goal of PCA:** capture most of the variation in original data set to bring out patterns.
 - **“fewer dimensions”** => reducing dimensionality and hence the curse of dimensionality
 - Basically we **compress the data set**.
 - Note: might lose some of variation of original data, and hence can't perfectly reproduce original data in the new subspace, but this variation is not important (due to being very small/insignificant).
 - This **new “subspace” comprises the Principal Components**
 - **IMPORTANT: PCA works only with numerical vectors!**

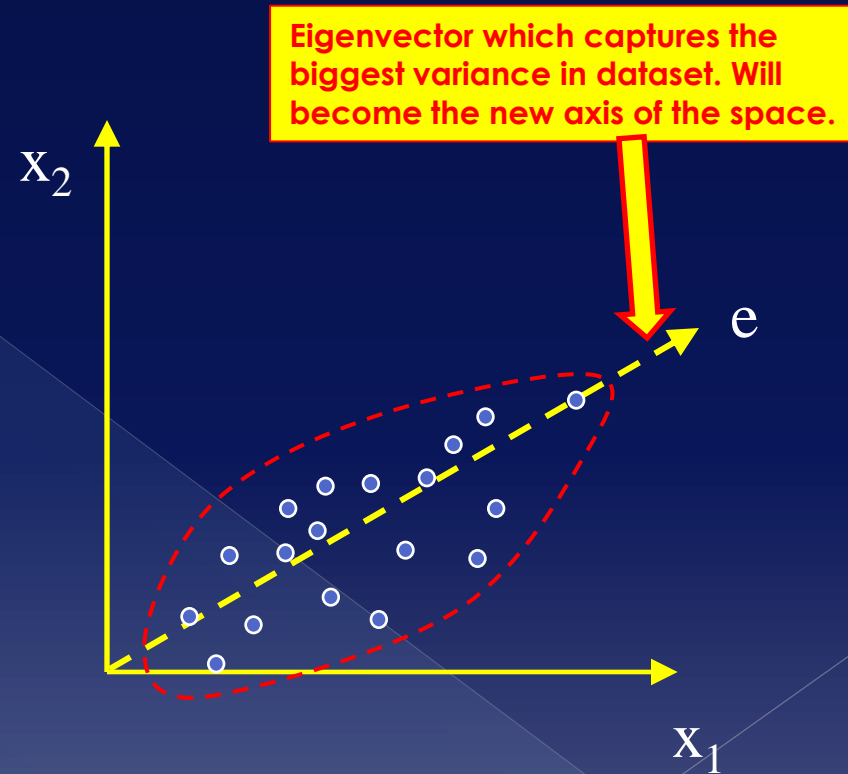
Principal Components Analysis (PCA)

- ◎ A quick look at PCA
 - > Several issues with the new subspace
 - How to choose new dimension d ?
 - How to select **feature space** (“subspace”) that represents our data well (i.e. principal components)?
 - > PCA allows you to create a new space (new variables) with fewer dimensions, which explains variation of the original dataset. You can then map your original data onto this new space and do your analysis there!

Principal Components Analysis (PCA)

- How to **find the axes (i.e. variables) of new space?**

- > The **Eigenvectors, Eigenvalues** of the Covariance matrix define these spaces
 - Eigenvectors are linear independent – i.e. orthogonal to each other
- > The **calculated Eigenvectors** (aka **Principal Components**) will be the **new axes** that define the new space upon which the data will be mapped.



Principal Components Analysis (PCA)

- ◉ What are Eigenvectors/Eigenvalues?
 - > Mathematical definition:
 - Let **A be a nxn matrix**. An **eigenvector v** and an **eigenvalue λ** of matrix A have the following properties:

$$A * v = \lambda * v$$

Principal Components Analysis (PCA)

- PCA is a method for finding the Eigenvectors and Eigenvalues of a dataset and use these vectors (**or a subset of these – the most important ones**) to create a new space with smaller dimensions, upon which the original data will be mapped while maintaining the variance of the dataset.
- You can then do your analysis (any analysis) **in this new space** which has fewer dimensions (variables) and move the data back and forth.

Principal Components Analysis (PCA)

- Steps to calculate Principal Components
 - > Take whole dataset with n dimensions
 - > Normalize the data – make your variable the to have the same SCALE!
 - Not always necessary if scale is not an issue with your data
 - > Compute the dimensional mean-vector (i.e. mean for each dimension/attribute)
 - > Subtract mean from each dimension (make variables have mean =0) – a form of **normalizing the data (THIS IS IMPORTANT!)**
 - > Compute the covariance matrix
 - Indicating how each dimension/attribute varies with respect to all other attributes
 - > Compute Eigenvectors and Eigenvalues of the covariance matrix solving:
 - **$|\lambda I - A| = 0$, λ =eigenvalue, $|\cdot|$ = determinant, I = unit vector**
 - **$Av = \lambda v$, v = eigenvector**
 - > Choose k largest Eigenvalues and corresponding Eigenvectors
 - > Use these Eigenvectors to form a d x k new matrix W of Eigenvectors – These **explain most of the variance**
 - > Use this d x k Eigenvector matrix to transform each object (vector) onto the new space, as follows:
 - **$\langle \text{New vector} \rangle = W^T x \langle \text{old_vector} \rangle$**


Data preprocessing

PCA Example

Principal Components Analysis (PCA) - Example

- Numerical Example
- Assume the following observations/data about different food items: vitamin C content, protein content)

Variables/features



Vitamin C	Protein
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

Records/Observations

Dataset

- Our goal/question to answer?
- **Reduce the number of variables while at the same time keep/explain most of the variance. Here e.g. we want to have only 1 variable**
 - > We have here only 2 variables, so this makes little sense. Imagine e.g. having 250 variables. In such cases you want to reduce the number of variables but “keep” the variance.

Principal Components Analysis (PCA) - Example

- Calculate mean for each variable

Vitamin C	Protein
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

mean = 1.81

mean = 1.91

- Scaling our variables not an issue here!
 - > **It would be an issue, if our variables would be measured in different scales. Say one variable is measured in thousands (e.g. population/income) and another in meters (say height of people). In such cases PCA will result in inconsistent results!**
 - > **When scale of any variable is an issue, do min-max normalization or z-score. Will see these methods in detail later.**

Principal Components

Analysis (PCA) - Example

- Subtract mean from each value to normalize the data.

Vitamin C	Protein
$2.5 - 1.81 = \mathbf{0.69}$	$2.4 - 1.91 = \mathbf{0.49}$
$0.5 - 1.81 = \mathbf{-1.31}$	$0.7 - 1.91 = \mathbf{-1.21}$
$2.2 - 1.81 = \mathbf{0.39}$	$2.9 - 1.91 = \mathbf{0.99}$
$1.9 - 1.81 = \mathbf{0.09}$	$2.2 - 1.91 = \mathbf{0.29}$
$3.1 - 1.81 = \mathbf{1.29}$	$3.0 - 1.91 = \mathbf{1.09}$
$2.3 - 1.81 = \mathbf{0.49}$	$2.7 - 1.91 = \mathbf{0.79}$
$2 - 1.81 = \mathbf{0.19}$	$1.6 - 1.91 = \mathbf{-0.31}$
$1 - 1.81 = \mathbf{-0.81}$	$1.1 - 1.91 = \mathbf{-0.81}$
$1.5 - 1.81 = \mathbf{-0.31}$	$1.6 - 1.91 = \mathbf{-0.31}$
$1.1 - 1.81 = \mathbf{-0.71}$	$0.9 - 1.91 = \mathbf{-1.01}$

mean = 1.81

mean = 1.91


NOTE: we will work from now on with the red values!

Hint: mean of both variables is now 0.

Principal Components Analysis (PCA) - Example

- Calculate the covariance matrix

	Vitamin C	Protein
Vitamin C	0.616	0.615
Protein	0.615	0.716

$$\text{cov}(\text{VitaminC}, \text{Protein}) = \frac{\sum_{i=1}^{10} (\text{VitC}_i - \overline{\text{VitC}})(\text{Prot}_i - \overline{\text{Prot}})}{9 = (10 - 1)}$$


NOTE: mean here is calculated on the normalized data (i.,e. mean = 0)

Principal Components Analysis (PCA) - Example

- Calculate **Eigenvalues and Eigenvectors of the Covariance matrix**
- Definition of Eigenvector v with Eigenvalue λ of the covariance matrix $\text{cov}(VitC, Pr)$:
 - > $\text{cov}(VitC, Pr)v = \lambda v \Rightarrow \text{cov}(Vit, Pr)v - \lambda v = 0 \Rightarrow$
 $(\text{cov}(Vit, Pr) - \lambda I_2)v = 0$
- Calculate **Eigenvalues first!**

Principal Components Analysis (PCA) - Example

- Calculate Eigenvalues first

$$(\mathit{cov}(V\mathit{it}, Pr) - \lambda I_2) v = 0$$



Note: 0 is the zero vector. We search for λ (eigenvalue) and corresponding v (eigenvector). Let's remember a little bit of linear algebra: In order for this to have non-zero vector v as solution, the determinant of $(\mathit{cov}(V\mathit{it}, Pr) - \lambda I_2)$ must be zero! Let's do it.

Principal Components Analysis (PCA) - Example

- Calculate Eigenvalues

$$\det(\text{cov}(Vit, Pr) - \lambda I_2) = 0$$

Covariance matrix

0.616	0.615
0.615	0.716

Identity matrix I_2

1	0
0	1

- λ

=

=

$0.616 - \lambda$	0.615
0.615	$0.716 - \lambda$



Determinant of this must be zero.

Principal Components Analysis (PCA) - Example

● Calculate Eigenvalues

$$\text{Det} \left(\begin{array}{|c|c|} \hline 0.616 - \lambda & 0.615 \\ \hline 0.615 & 0.716 - \lambda \\ \hline \end{array} \right) = 0 \Rightarrow$$

$$\Rightarrow (0.616 - \lambda)(0.716 - \lambda) - 0.615 * 0.615 = 0 \Rightarrow \lambda_1 = 0.0489, \lambda_2 = 1.283$$

2 Eigenvalues calculated λ_1, λ_2 !

Principal Components Analysis (PCA) - Example

- Now, **for each Eigenvalue**, calculate the **Eigenvector V**.

For eigenvalue $\lambda = 0.0490$

$0.616 - \lambda$	0.615
0.615	$0.716 - \lambda$

$$* \mathbf{V} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow$$

$0.616 - 0.0489$ $= 0.567$	0.615
0.615	$0.716 - 0.0489$ $= 0.667$

$$* \mathbf{V} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow$$

Principal Components Analysis (PCA) - Example

- For each Eigenvalue, calculate the Eigenvectors.

For eigenvalue $\lambda = 0.0490$

0.567	0.615
0.615	0.667

$$\begin{matrix} * & \begin{matrix} v1 \\ v2 \end{matrix} & = & \begin{matrix} 0 \\ 0 \end{matrix} & \Rightarrow \end{matrix}$$

$$\begin{cases} 0.567*v1 + 0.615*v2 = 0 \\ 0.615*v1 + 0.667*v2 = 0 \end{cases}$$



Eigenvalue $\lambda = 0.049$

$$\text{Eigenvector} = \begin{bmatrix} -0.7351 \\ 0.6778 \end{bmatrix}$$

IMPORTANT! This system of equations has an infinite number of solutions (which makes sense).

Principal Components Analysis (PCA) - Example

- For each Eigenvalue, calculate the Eigenvectors.

For Eigenvalue $\lambda = 1.284$

$0.616 - \lambda$	0.615
0.615	$0.716 - \lambda$

$$* \mathbf{V} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow$$

$0.616 - 1.284 =$ -0.668	0.615
0.615	$0.716 - 1.284 =$ -0.568

$$* \mathbf{V} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow$$

Principal Components Analysis (PCA) - Example

- For each Eigenvalue, calculate the Eigenvectors.

For Eigenvalue $\lambda = 1.284$

-0.668	0.615
0.615	-0.568

 *

v1
v2

 =

0
0

 =>

$$\begin{aligned} -0.668*v1 + 0.615*v2 &= 0 \\ 0.615*v1 + 0.568*v2 &= 0 \end{aligned}$$

Eigenvalue $\lambda = 1.284$

Eigenvector = $\begin{bmatrix} -0.6778 \\ -0.7351 \end{bmatrix}$

Principal Components Analysis (PCA) - Example

Eigenvalue $\lambda=1.284$

Eigenvector = $\begin{bmatrix} -0.6778 \\ -0.7351 \end{bmatrix}$

Eigenvalue $\lambda=0.049$

Eigenvector = $\begin{bmatrix} -0.7351 \\ 0.6778 \end{bmatrix}$

- These two eigenvectors define a new coordinate system upon which the original data can be projected.
- What variables do these represent?
 - > Define **new variables** as linear combinations of the initial variables i.e.:
 - **New variable 1 = $-0.6778 \cdot \text{VitaminC} - 0.7351 \cdot \text{Protein}$**
 - **New variable 2 = $-0.7351 \cdot \text{VitaminC} + 0.6778 \cdot \text{Protein}$**

Principal Components Analysis (PCA) - Example

- We found **2 Eigenvalue/Eigenvector** pairs (that's expected. Why expected?)

Eigenvalue $\lambda=1.284$

Eigenvector = $\begin{bmatrix} -0.6778 \\ -0.7351 \end{bmatrix}$

Eigenvalue $\lambda=0.049$

Eigenvector = $\begin{bmatrix} -0.7351 \\ 0.6778 \end{bmatrix}$

- Notice how **one Eigenvalue is greater than the other ? $1.284 > 0.049$.**
 - > This means that **the Eigenvector with $\lambda = 1.284$ captures more variance of the dataset than the other Eigenvector! And in fact, the value of the Eigenvalue is the variance of the data on that (new) dimension!**

Principal Components Analysis (PCA) - Example

- **How much variance** does the greatest Eigenvector explain?
 - > Use $\frac{\lambda_k}{\sum_{i=1}^n \lambda_i}$ where n= number of eigenvalues/eigenvectors - to see how much variance Eigenvector with Eigenvalue λ_k explains
 - > In our case Eigenvector with $\lambda=1.284$ explains **1.284 / (1.284+0.049) = 0.96** or **96% of the variance** in the data
 - > In the general case, what you do is **select the k largest Eigenvalues (and corresp. Eigenvectors)** until you are happy with the variance explained – **The selected Eigenvalues/Eigenvectors are the Principal Components!**
 - In this case the explained variance is $\frac{\sum_{i=1}^k \lambda_i}{\sum_{j=1}^n \lambda_j}$
 - Empirical: **aiming at explaining >70% or variance**

Principal Components Analysis (PCA) - Example

- If we are happy with the variance explained, do the following:
 - > Map the original data to the selected k Eigenvectors with the k greatest eigenvalues -in our example, lets say we select only 1 Eigenvalue/Eigenvector pair – the one with the largest Eigenvalue :

Vitamin C	Protein
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



Map this data to this feature vector defined by the Eigenvector

Eigenvalue $\lambda=1.284$

Eigenvector = $\begin{bmatrix} -0.6778 \\ -0.7351 \end{bmatrix}$

We aim to do this => Express data solely in terms of the selected Eigenvectors that define a new space!

Note: The two dimensional, original data will be mapped to a one dimensional space explaining large part of the variance!

Principal Components Analysis (PCA) - Example

- What variable(s) do the Eigenvectors (which defined the new space) represent?
 - > It's a **new variable** not in the original dataset! In fact a linear combination of existing variables.

VitaminC

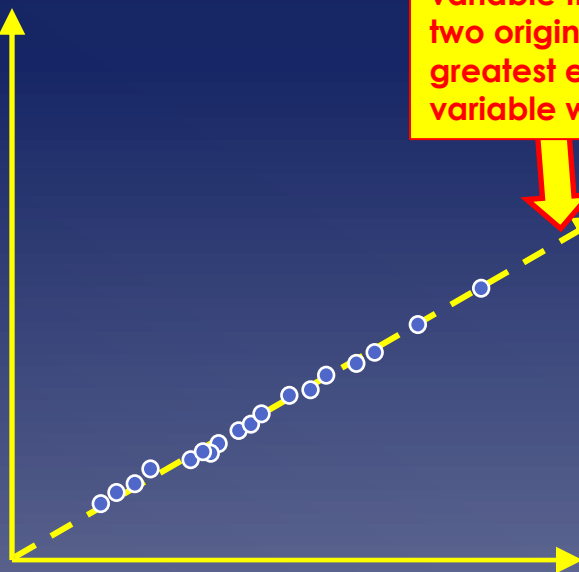
Here we have reduced the two variables (VitaminC, Protein) to only one variable represented by the Eigenvector, and mapped the existing dataset onto that new axis (see the blue points)

Which variable is this?????

None of the existing ones in the dataset! it's a new variable that is created from a linear combination of the two original variables: If this is the eigenvector with the greatest eigenvalue (explaining most of the variance) this variable would be $-0.6778 \cdot \text{VitaminC} - 0.7351 \cdot \text{Protein}$

Eigenvector $\begin{bmatrix} -0.6778 \\ -0.7351 \end{bmatrix}$

Protein



Principal Components Analysis (PCA)

- ◎ Summary
- ◎ Principal Component Analysis - PCA
 - > **What it does**
 - It reduces the number of dimensions in a dataset, while still explaining great amount of variance in the original data
 - > **On what kind of attributes/data does it work?**
 - PCA works **only on Ratio attributes**.
 - Variations of PCA to work in interval data available.
 - > **PCA can make use of the Correlation matrix instead of the Covariance matrix**
 - Important when implementing PCA in R
 - Look at the appropriate parameters!

Principal Components Analysis (PCA)

Principal Component Analysis – PCA

> **When to do it**

- When you want to visualize datasets with many variables (e.g. > 100)
 - Impossible to do with that many variables
 - With PCA, keep e.g. the 3 principal eigenvectors and project data onto that space. 3 variables can be visualized easily.
- Identify correlated variables in datasets with many variables.

Principal Components Analysis (PCA)

- A more concrete example:
- *"In the Places Rated Almanac, Boyer and Savageau rated 329 communities according to the following nine criteria:*
 - > *Climate and Terrain*
 - > *Housing*
 - > *Health Care & the Environment*
 - > *Crime*
 - > *Transportation*
 - > *Education*
 - > *The Arts*
 - > *Recreation*
 - > *Income*
 - > ...

Note that within the dataset, except for housing and crime, the higher the score the better. For housing and crime, the lower the score the better. Where some communities might do better in the arts, other communities might be rated better in other areas such as having a lower crime rate and good educational opportunities."

Objective: Search for relationships (correlation) between these variables.

Principal Components Analysis (PCA)

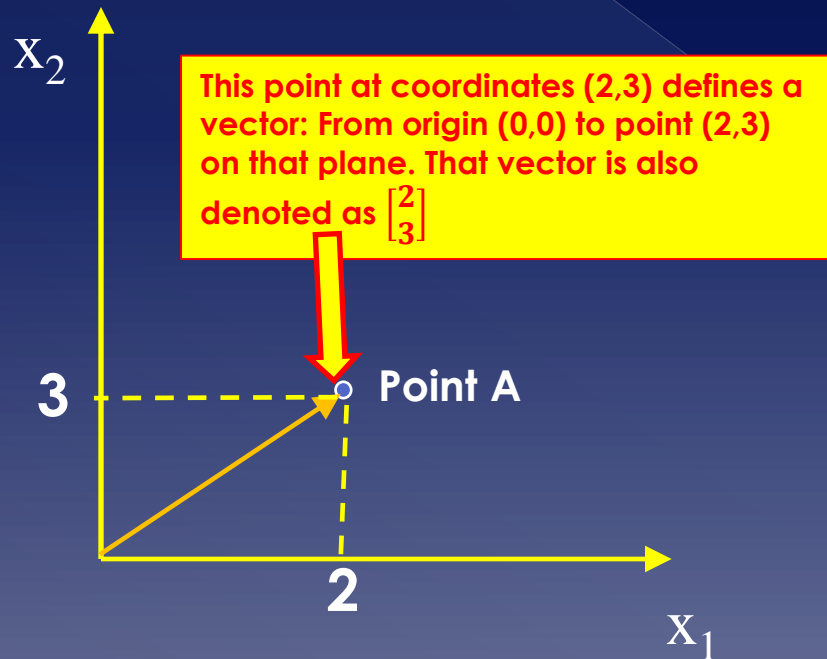
- In order to do this you need to check all combination of variables and expect a linear correlation
 - > In this 9-dimensional space, observation which are correlated will appear closely together
 - > **Difficult to see**: too many scatterplots of variables against each other, how to draw a 9-dimensional space etc.
- Or you could **do a PCA**, find principal components and project data onto these
 - > Such projection **gives you a quick view of the grouping** which implies correlation.

Data preprocessing

PCA – Sorry, I still don't get it!

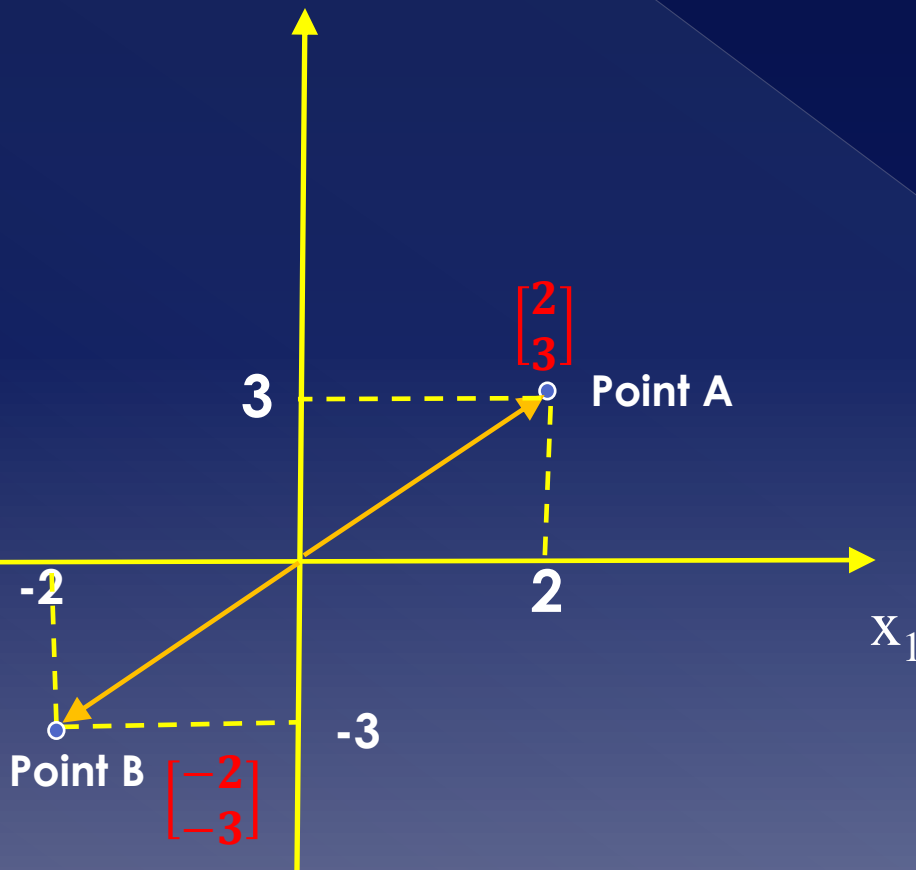
Principal Components Analysis (PCA) – I don't get it

- I'm sorry, I still don't get PCA. Get you draw it for me? **Ok, first some basics about matrix multiplication.**



Principal Components Analysis (PCA)

- Notice how point B is a reflection of Point A on the origin (0,0)?



The question is now: how can we calculate the reflection of any point P in the origin?

Easy: Just multiply the vector with the matrix $\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$ this will calculate the vector that is the reflection of the original e.g.

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} -2 \\ -3 \end{bmatrix}$$

From this, please take away the following important message: Matrix multiplication is simply Vector TRANSFORMATIONS (i.e. move vector elsewhere)! Note: you can define matrices for any transformation. If you multiply matrices A and B with vector $\begin{bmatrix} -2 \\ -3 \end{bmatrix}$ i.e. $A*B*\begin{bmatrix} -2 \\ -3 \end{bmatrix}$ that means: transform vector $\begin{bmatrix} -2 \\ -3 \end{bmatrix}$ according to B and the result according to A. This may indicate e.g. Rotate and Mirror vector.

Principal Components Analysis (PCA)

- Multiplication of matrices are transformations

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$



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



Imagine this matrix as a function f transforming vector $(2,3) \rightarrow$ reflection over x

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$



Function or Filter for reflecting over y axis

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$



Function or Filter for reflecting over $y=x$ line

Principal Components Analysis (PCA)

- May also transform onto lower dimensional spaces

$$\underbrace{[2 \quad 7]}_{\text{Function transforming vector } [2 \ 3]} \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$= 25$$



This transformation $[2 \ 7]$ maps a vector onto a line i.e. one dimensional space.

Function
transforming vector
 $[2 \ 3]$

Principal Components Analysis (PCA)

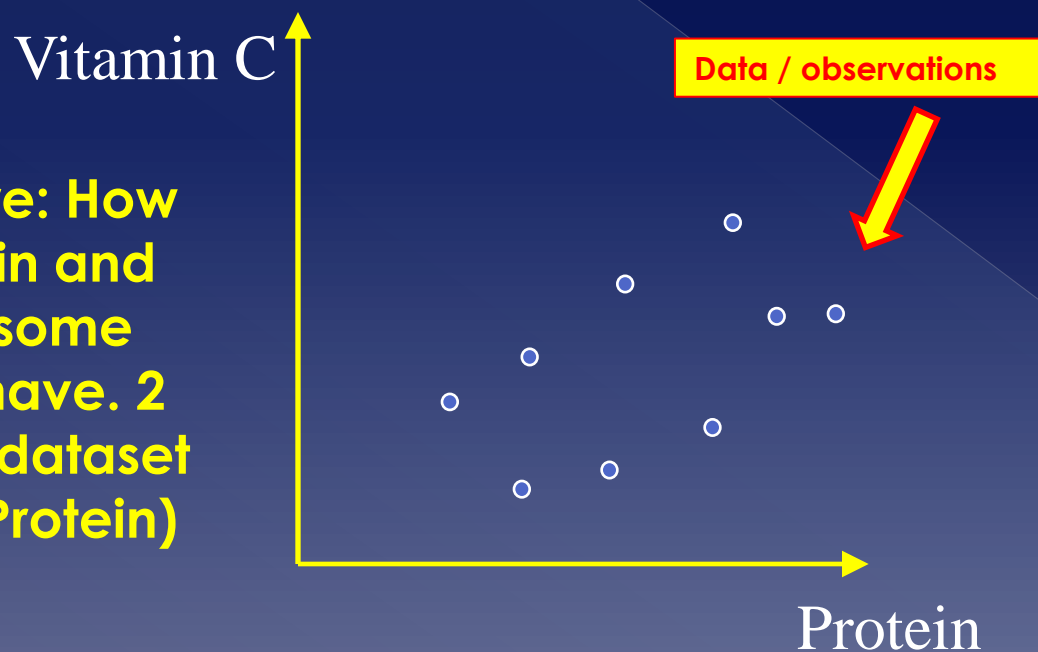
- Matrix multiplication is not multiplication!
It's transformation i.e. moving a point from one position on the same axes to another position, or moving a point from the current axis system to a new/different axis system.

Data preprocessing

PCA – A visual explanation

Principal Components Analysis (PCA)

- Let's assume we have some data with 2 variables (2 dimensions) and we want to reduce the number of variables to 1 while **keeping the variance, as much as possible, of the original data.**

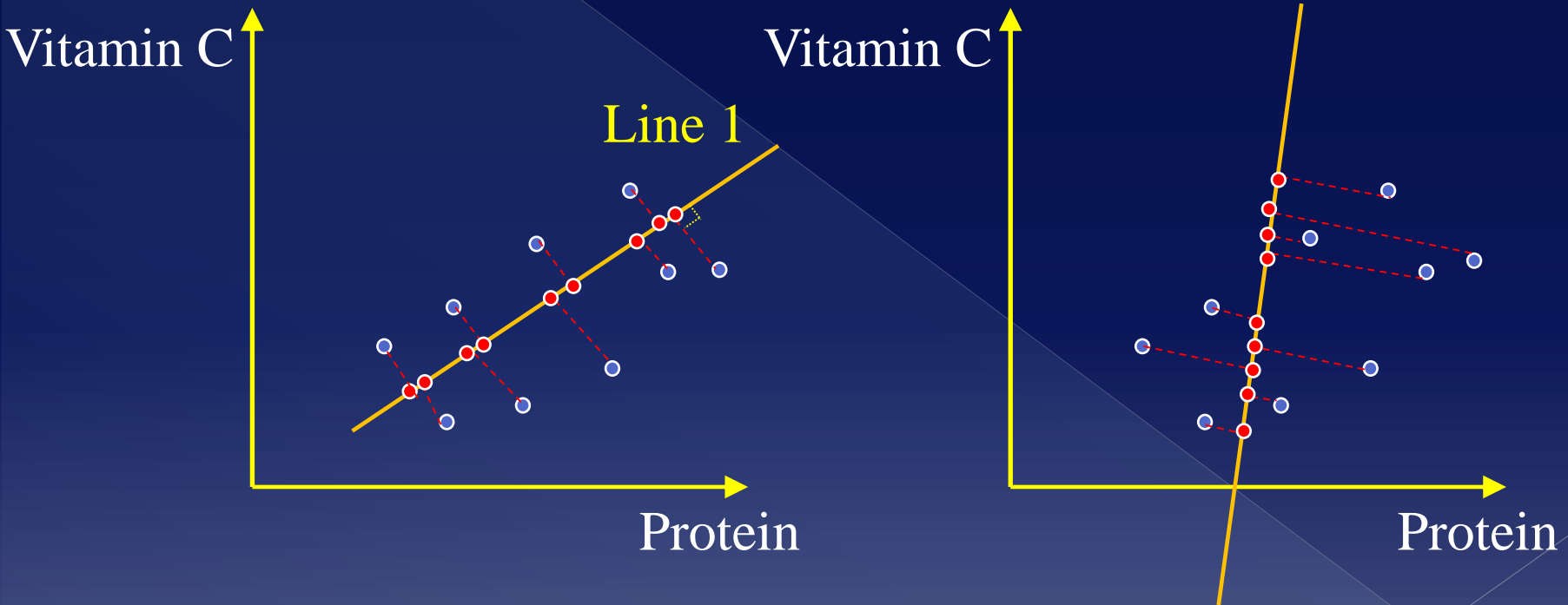


Example here: How much Protein and Vitamin C some food items have. 2 dimensional dataset (Vitamin C, Protein)

Principal Components Analysis (PCA)

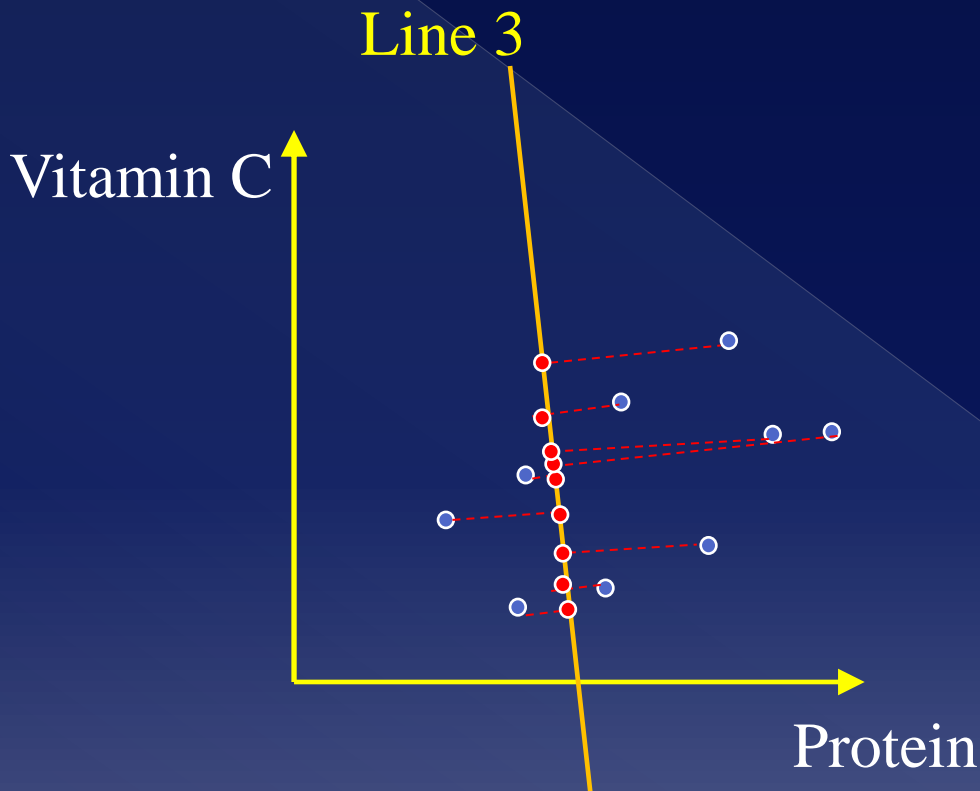
- Let's do the following now: **Draw random lines** on the plane of your data **and project the original data on that line. This would mean in essence projecting my 2-dimensional data on a 1-dimensional space.** How does this look like?
 - > This **line will be the new axis** upon which the 2-dimensional data will be projected and become 1-dimensional
 - > Our goal is to explain/keep most of the variance of the original dataset when the data is projected onto the new space. (**That's what PCA does**)

Principal Components Analysis (PCA)



Note: Red dots on Line 1 and Line 2 are the projections of the 2-dimensional data on each line (1-dimensional space). Projections are perpendicular to the lines. HINT: Notice how the “spread” (aka variance) of the red dots on these lines (Line1, Line2) differ?

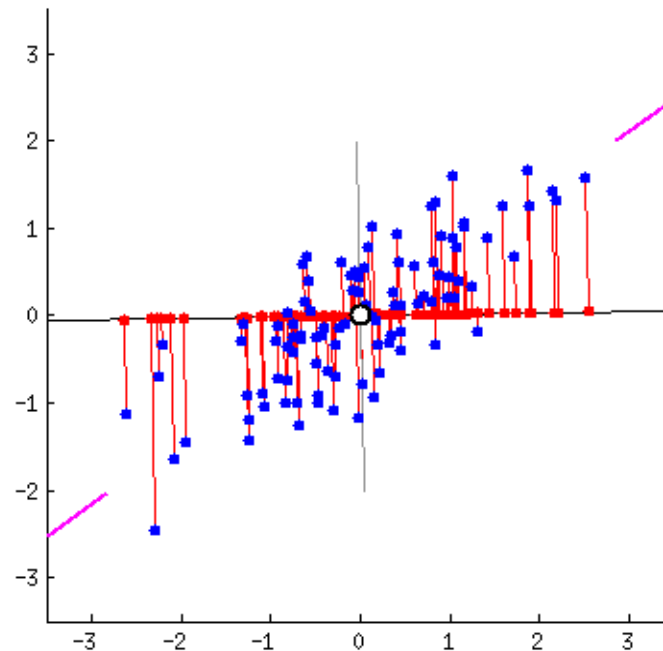
Principal Components Analysis (PCA)



Compare the spacing of the red dots (variance) on Line 3 to Line 1 and 2. See how the spread of red dots on Line 3 is well.... Smaller than on Lines 2 and 3? That means that Line 3 captures a smaller variance of the original dataset!

Principal Components Analysis (PCA)

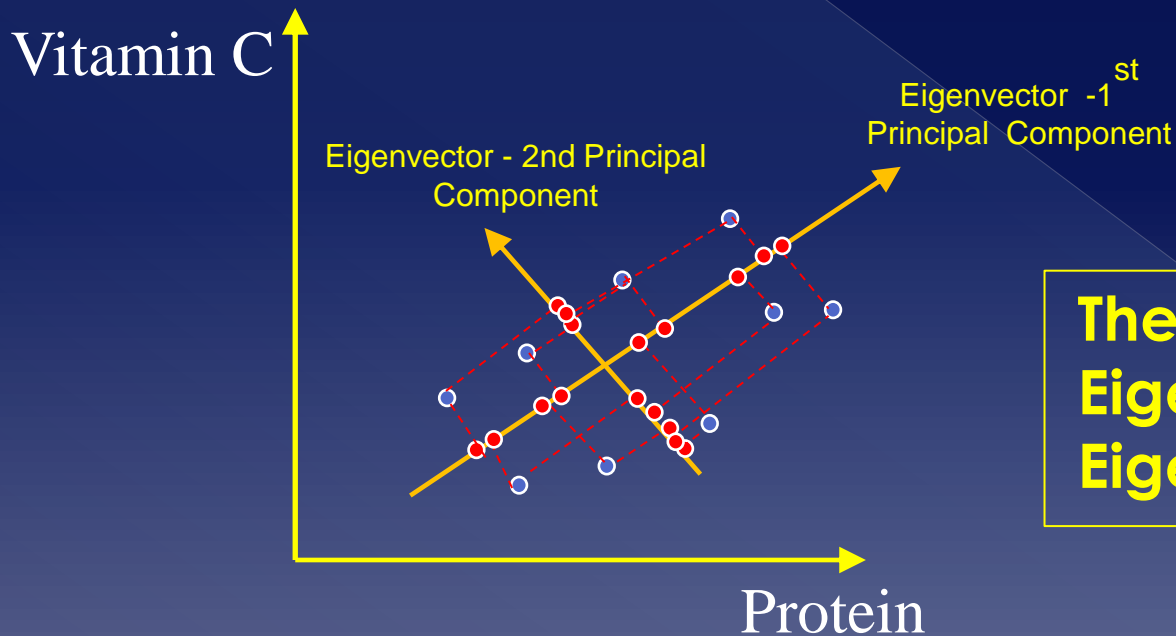
You can draw indefinitely many such lines (see rotating line) and project the data onto them. On some lines, the “spread” of red dots i.e. variance of red dots on the line will be greater than on others. These lines (or vectors) are the Eigenvectors!



For animated version see file: PCA-Eigenvector-Illustration.gif

Principal Components Analysis (PCA)

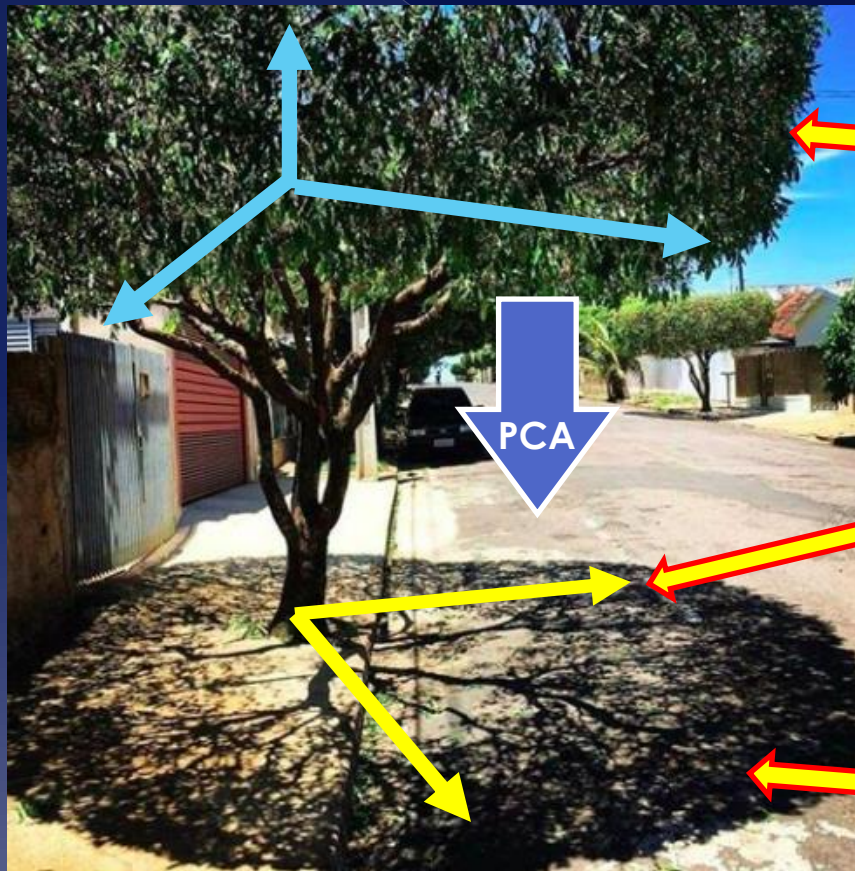
- The line where the red dots have **the greatest variance (biggest spread)**, is an **Eigenvector** and the **First Principal Component** of our data! The **variance (spread)** of red dots are the **Eigenvalues of the Eigenvectors!**
 - > The line with **the second biggest spread** is the **second Principal Component**, the line with the **third biggest spread** is the **Third Principal Component** etc.



The length of the Eigenvector is its Eigenvalue λ

Principal Components Analysis (PCA)

- A metaphor describing what PCA does



Original data / observations (here represented as leaves and branches) in 3 dimensional space i.e. original data has 3 features

Eigenvectors defining a new space upon which data is projected. Not all Eigenvectors need to be used to define the new space. Here only 2 are used to define the new space.

Data / observations (leaves and branches) projected onto a new 2 dimensional space (shadows of leaves and branches) defined by 2 eigenvectors i.e. 2 features designed in such way so as to maximize the number of leaves and branches visible. This is achieved if the variance is maximized i.e. data is spaced out.

Principal Components Analysis (PCA)

- ⦿ Using **Eigenvalues/Eigenvectors** is **one way to do PCA**
- ⦿ **Other ways** also available
 - > E.g. **using Single Value Decomposition – SVD**
- ⦿ Both methods **yield to similar results**
 - > i.e. not much difference.

Principal Components Analysis (PCA)

PCA in R:

See file `PCA.R` on eclass performing PCA on the Iris dataset.

PCA in Python:

See file `PCA.py` on eclass performing PCA on the Iris dataset.

Discretization

- Discretization?
 - > **Divide the range of a continuous** attribute **into intervals**
 - > Some classification algorithms only accept categorical attributes.
 - > **Reduce data size by discretization**
 - > Prepare for further analysis
 - > Used in problems that require categorization and correlation analysis

Discretization

- ◎ Two ways to discretization

- > **Unsupervised**

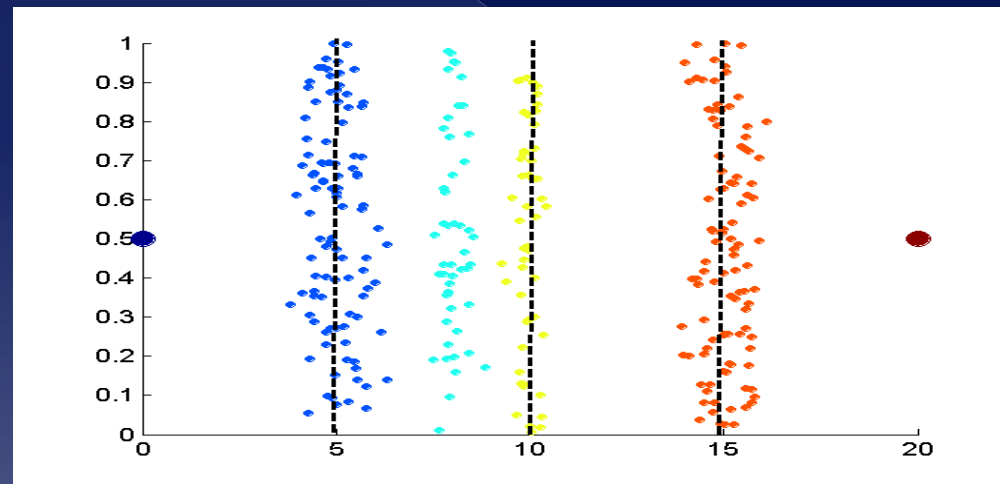
- Don't take into consideration the classes in which the data item belong

- > **Supervised**

- Take into consideration the classes in which data items belong

Discretization

- Unsupervised methods
 - > **Equal interval width** : Split range in **n equal spaces** by specifying **n-1 split points**

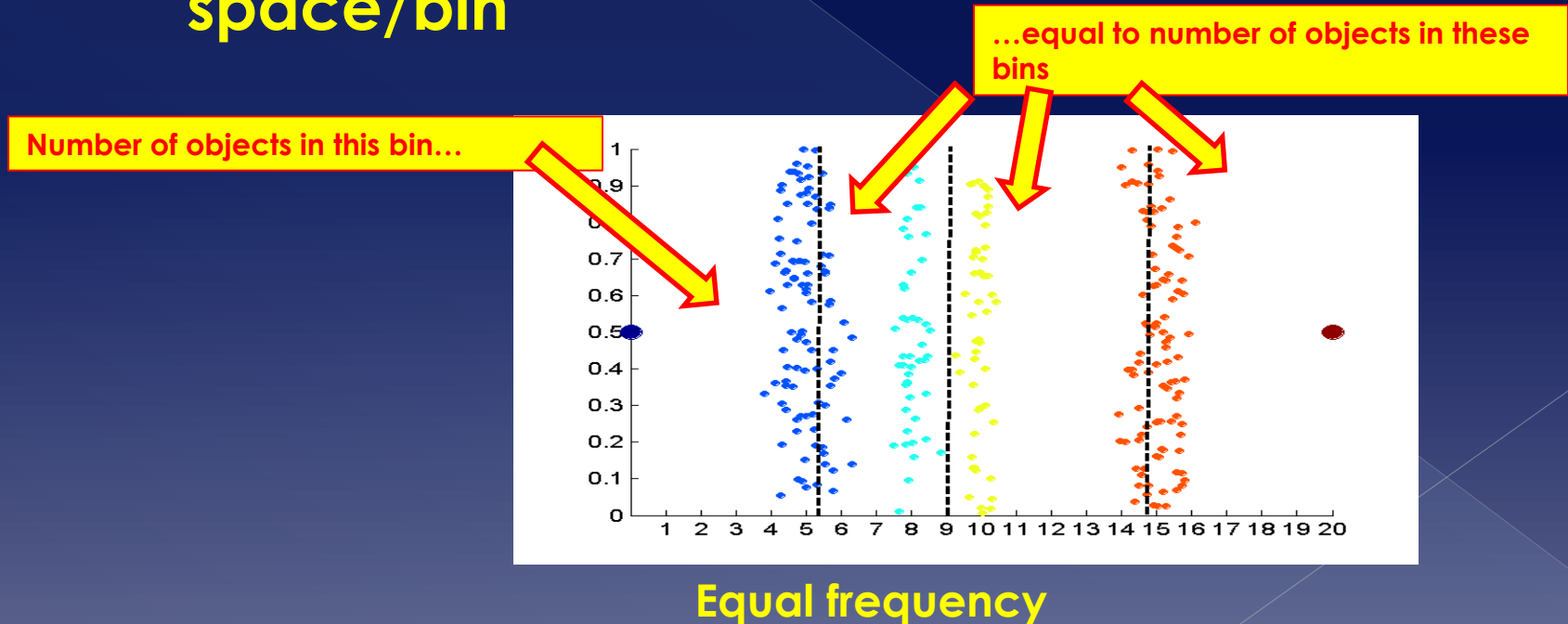


Equal interval width

Discretization

- Unsupervised methods

- > **Equal frequency**: Split range in spaces so **that equal number of data objects are in each space/bin**



Discretization

- Supervised methods
 - > Here **we look at some attribute (class)** of the data and try to take this into consideration when building the bins (hence supervised). Try to **improve quality of bins wrt class**.
 - > Bottom-up approach
 - Each item belongs to its own bin. Then try to produce bigger bins by evaluating some metrics
 - > Goal: create **bins that are as “clean” as possible wrt an attribute**, i.e. minimize “chaos”/”unorderly-ness” in each bin in terms of the class the items belong.

Discretization

- ◉ Can we **measure “chaos”/“unorderly-ness”** in each bin?
 - > Yup, that is what **Entropy** does
 - > Measuring entropy of bin e_i :

$$e_i = \sum_{j=1}^k \frac{m_{i,j}}{m_i} \log_2 \frac{m_{i,j}}{m_i}$$

...where k the number of different bins/classes, m_i the number of items in class i , $m_{i,j}$ the number of items that are in class j found in bin i . $m_{i,j}/m_i$ is the probability of class j in bin i .

Discretization

- Total entropy e of the spaces/partitioning is defined as

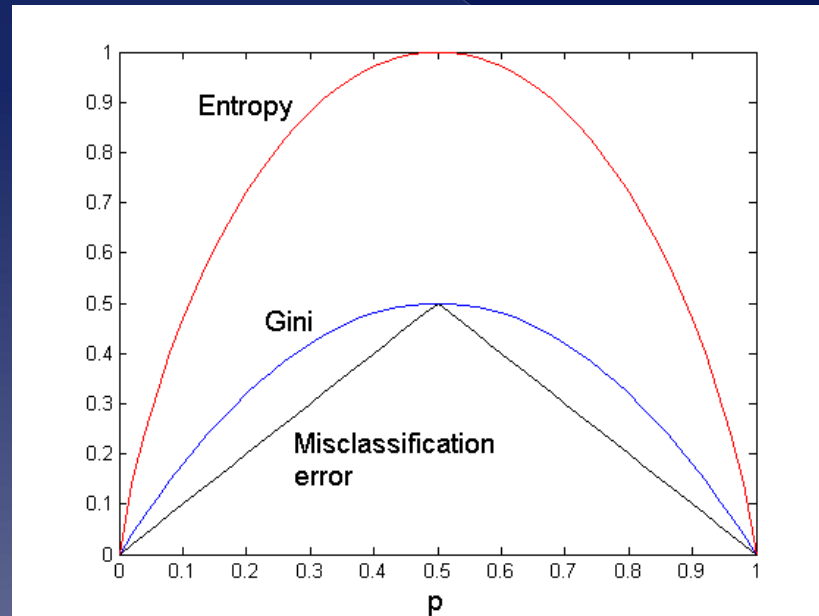
$$e = \sum_{i=1}^n \frac{m_i}{m} e_i$$

...where m total number of data items, m_i the number of data items in bin i (defined in terms of class)

Discretization

Some notes on Entropy

- > **If Entropy = 0** => no chaos, perfect order, clean space/partition. **Minimum entropy**
- > **If Entropy = 1** => biggest chaos, greatest “unorder”, most unclean space/partition. **Maximum entropy**



Similarity and dissimilarity measures

Similarity and dissimilarity measures

◎ **Similarity**

- > Numerical measure of how alike two data objects are.
- > Is higher when objects are more alike.
- > Often falls in the range $[0,1]$

◎ **Dissimilarity**

- > Numerical measure of how different two data objects are
- > Lower when objects are more alike
- > Minimum dissimilarity is often 0
- > Upper limit varies

◎ **Proximity** refers to a similarity or dissimilarity

Similarity and dissimilarity measures

- For simple attributes
 - Note: q, p below are attribute values for two data objects**
 - s, d below stand for (s)imilarity and d(istance)**

Attribute Type	Dissimilarity	Similarity
Nominal	$d = \begin{cases} 0 & \text{if } p = q \\ 1 & \text{if } p \neq q \end{cases}$	$s = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases}$
Ordinal	$d = \frac{ p-q }{n-1}$ <p>(values mapped to integers 0 to $n-1$, where n is the number of values)</p>	$s = 1 - \frac{ p-q }{n-1}$
Interval or Ratio	$d = p - q $	$s = -d, s = \frac{1}{1+d} \text{ or } s = 1 - \frac{d - \min_d}{\max_d - \min_d}$

Table 5.1. Similarity and dissimilarity for simple attributes

Similarity and dissimilarity measures

Distance

- > is an **dissimilarity measure**
- > Observe that **Dissimilarity and Distance** are same things
 - You use distance to measure similarity/dissimilarity
 - You **transform distance** in order to calculate similarity/dissimilarity e.g. **similarity** = $\frac{1}{\text{distance}(p_1, p_w)}$ **or**

similarity = $\frac{1}{e^{\text{distance}(p_1, p_2)}}$, etc. In general you choose the proper formula.

Different ways to measure distance

- > **Euclidian** distance
- > **Minkowski** distance
- > **Mahalanobis** distance

Similarity and dissimilarity measures

- You can define **your own distance measure**.
- However, in order to be considered a proper distance measure, it must be a metric. Or more clearly **it has to have the following properties:**

1. $d(x, y) \geq 0$

2. $d(x, y) = 0$ iff $x = y$

3. $d(x, y) = d(y, x)$

4. $d(x, z) \leq d(x, y) + d(y, z)$

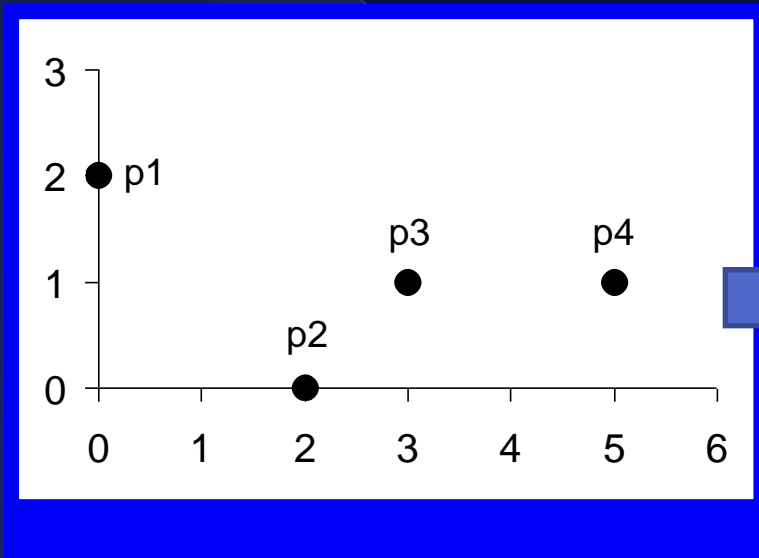
Euclidian Distance

- **“Works” for points x , y in one, two, three or more dimensions**
- (known) Formula

$$d(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2}$$

...where n is the number of dimensions (attributes) and x_k and y_k are, respectively, the k^{th} attributes (components) or data objects x and y .

Euclidian Distance



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

Data objects

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Euclidean Distance Matrix

The problem with the Euclidean distance

○ E.g.

- > $Euclidean([0.8, 1.2, 2.6], [0.1, 0.9, 2.1]) = 0.9119$
- > $Euclidean([0.8, 80000, 2.6], [0.1, 67090, 2.1]) = 12910.0002$

Large numbers influence more the Euclidean distance than small numbers. Here, Euclidean distance approx. equal to $|80000 - 67090| = 12910$ so why spend such calculation cost (raising power, sqrt)?

The problem with the Euclidean distance

- Solution: Normalizing (all or some) numbers that are at different scales (Feature scaling)
 - > Normalizing: Scaling into a fixed range 0-1
 - > Various approaches:
 - Min-max normalization of all values in dimension

$$\text{New value} = \frac{\text{current value} - \text{min value}}{\text{max value} - \text{min value}}$$

The problem with the Euclidean distance

- ◉ Min-max normalization (feature scaling)

- > $Euclidean([0.8, 80000, 2.6], [0.1, 67090, 2.1])$
- > Scaling 80000 and 67090 (belong to same dimension)

- $New\ value\ 80000 = \frac{80000 - 67090}{80000 - 67090} = 1$

- $New\ value\ 67090 = \frac{67090 - 67090}{80000 - 67090} = 0$

$$Euclidean([0.8, 1, 2.6], [0.1, 0, 2.1]) = 1.319$$

Much more sensible value than 12910

Minkowski distance

- ◉ Minkowski distance **is a generalization of Euclidean Distance**

$$d(x, y) = \sqrt[r]{\sum_{k=1}^n |x_k - y_k|^r}$$

...where r is a parameter, n is the number of dimensions (attributes) and x_k and y_k are, respectively, the **k th attributes** (components) or **data objects x and y** .

Minkowski distance

- **Special cases of the Minkowski distance:**
- **$r = 1$.** City block (Manhattan distance, taxicab, L_1 norm) distance.
 - A common example of this is the **Hamming distance**, which is just the number of bits that are different between two binary vectors
- **$r = 2$.** Euclidean distance
- **$r \rightarrow \infty$.** “Supremum” (L_{\max} norm, L_{∞} norm) distance.
 - This is the maximum difference between any component of the vectors
- **IMPORTANT!** Do not confuse r with n , i.e., all these distances are defined for all numbers of dimensions.

Minkowski distance

Manhattan distance, $r=1$



L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

Euclidean distance, $r=2$



L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance $r \rightarrow \infty$,
 L_∞ norm



L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

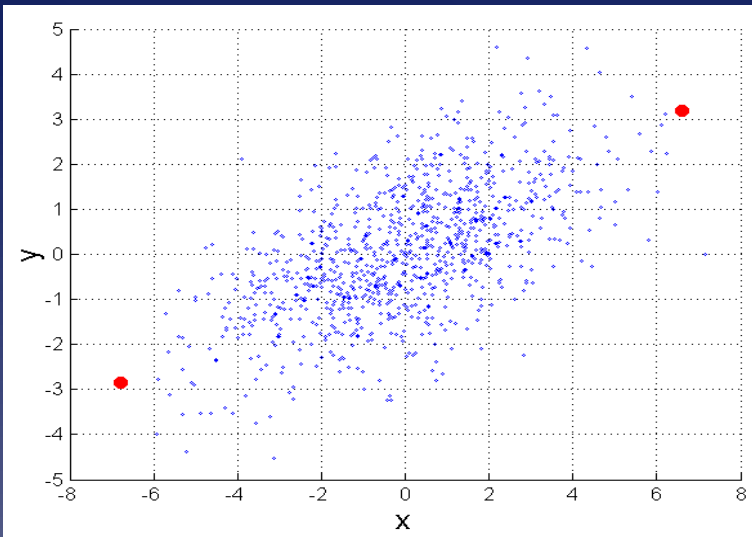
point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

Distance Matrices

Mahalanobis distance

- Is the distance **between a point p and a distribution D**
 - If **Mahalanobis distance = 0**, then point is at the mean of D (i.e. the “center”)

For red points, the Euclidean distance is 14.7, Mahalanobis distance is 6.



Mahalanobis distance of point p

$$\text{from distribution} = \sqrt{\sum_{i=1}^d \left(\frac{p_i - c_i}{\sigma_i} \right)^2}$$

p_i value of point in i dimension, c_i value of distribution center at dimension i , σ_i stdev of dimension i

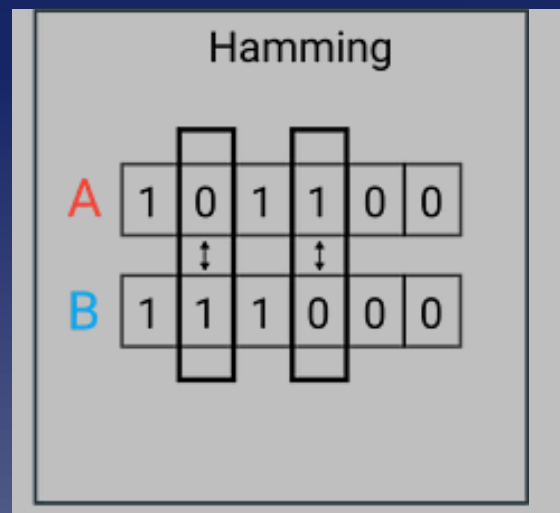
$$d(x, y) = (p - q)\Sigma^{-1}(p - q)^T$$

Mahalanobis distance

- Mahalanobis measures the distance in a multivariate space
- Mahalanobis distance equals the Euclidean distance of two points if all variables are **uncorrelated to each other** (orthogonal)
- Mahalanobis measures the distance if variables are correlated
 - > i.e. when axes **not in right angles**

Hamming distance

- ◉ Hamming distance
 - > between two vectors of equal length it's the number of positions at which the corresponding symbols are different.



Hamming distance = 2

Hamming distance

- Example of Hamming distance

Hamming Distance between nominal vectors:

$$d \left(\begin{array}{l} [Real\ Madrid, Blue, Transformers], \\ [Dortmund, Blue, 12\ Angry\ Men] \end{array} \right) = 2$$

Similarity between nominal vectors:

$$s \left(\begin{array}{l} [Real\ Madrid, Blue, Transformers], \\ [Dortmund, Blue, 12\ Angry\ Men] \end{array} \right) = \frac{1}{3}$$

Cosine similarity

- **Applies to document data**

- If d_1 and d_2 are two document vectors, then

$$\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1|| ||d_2||,$$

where \bullet indicates vector dot product and $||d||$ is the length of vector d .

- Example:

$$d_1 = (3, 2, 0, 5, 0, 0, 0, 2, 0, 0)$$

$$d_2 = (1, 0, 0, 0, 0, 0, 0, 1, 0, 2)$$

$$d_1 \bullet d_2 = 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5$$

$$||d_1|| = (3*3 + 2*2 + 0*0 + 5*5 + 0*0 + 0*0 + 0*0 + 2*2 + 0*0 + 0*0)^{0.5} = (42)^{0.5} = 6.481$$

$$||d_2|| = (1*1 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 0*0 + 1*1 + 0*0 + 2*2)^{0.5} = (6)^{0.5} = 2.245$$

$$\text{Hence, } \cos(d_1, d_2) = .3150$$

Jaccard index (Jaccard coefficient)

- Measures similarity between two sets
- Formula

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$



$$J(A, B) = \frac{1}{3}$$

$$\text{Jaccard distance} = 1 - J(A, B) = \frac{2}{3}$$

Summary

Summary

- Data has different types of attributes depending on the type of values they may take

Summary

- ◎ **Different types** imply **different methods of analysis**
 - > Different methods of analysis work for on different types of data
- ◎ **Preprocessing** is one of the most important steps in data mining
 - > Consumes most of the time (70-80% of dm tasks)
- ◎ There are **different objectives** when preprocessing data
 - > Reducing dimensions
 - > Sampling
 - > Discretization

Summary

- ◎ **PCA** most powerful way to reduce dimensions of the dataset (curse of dimensionality) which causes problems in **Big Data**
 - > Used in many-many Big Data environments
- ◎ There are also **different distance metrics**
 - > Depending on the data, objective of task at hand
- ◎ In general, **choose wisely, the appropriate, types of values, preprocessing methods and distance metrics**
 - > Will **influences** your data mining **results!**

Appendices

APPENDIX A: Related Bibliography

- D. P. Ballou and G. K. Tayi. Enhancing data quality in data warehouse environments. *Communications of ACM*, 42:73-78, 1999
- T. Dasu and T. Johnson. *Exploratory Data Mining and Data Cleaning*. John Wiley & Sons, 2003
- T. Dasu, T. Johnson, S. Muthukrishnan, V. Shkapenyuk. *Mining Database Structure; Or, How to Build a Data Quality Browser*. SIGMOD'02.
- H.V. Jagadish et al., Special Issue on Data Reduction Techniques. *Bulletin of the Technical Committee on Data Engineering*, 20(4), December 1997
- D. Pyle. *Data Preparation for Data Mining*. Morgan Kaufmann, 1999
- E. Rahm and H. H. Do. Data Cleaning: Problems and Current Approaches. *IEEE Bulletin of the Technical Committee on Data Engineering*. Vol.23, No.4
- V. Raman and J. Hellerstein. *Potters Wheel: An Interactive Framework for Data Cleaning and Transformation*, VLDB'2001
- T. Redman. *Data Quality: Management and Technology*. Bantam Books, 1992
- Y. Wand and R. Wang. Anchoring data quality dimensions ontological foundations. *Communications of ACM*, 39:86-95, 1996
- R. Wang, V. Storey, and C. Firth. A framework for analysis of data quality research. *IEEE Trans. Knowledge and Data Engineering*, 7:623-640, 1995