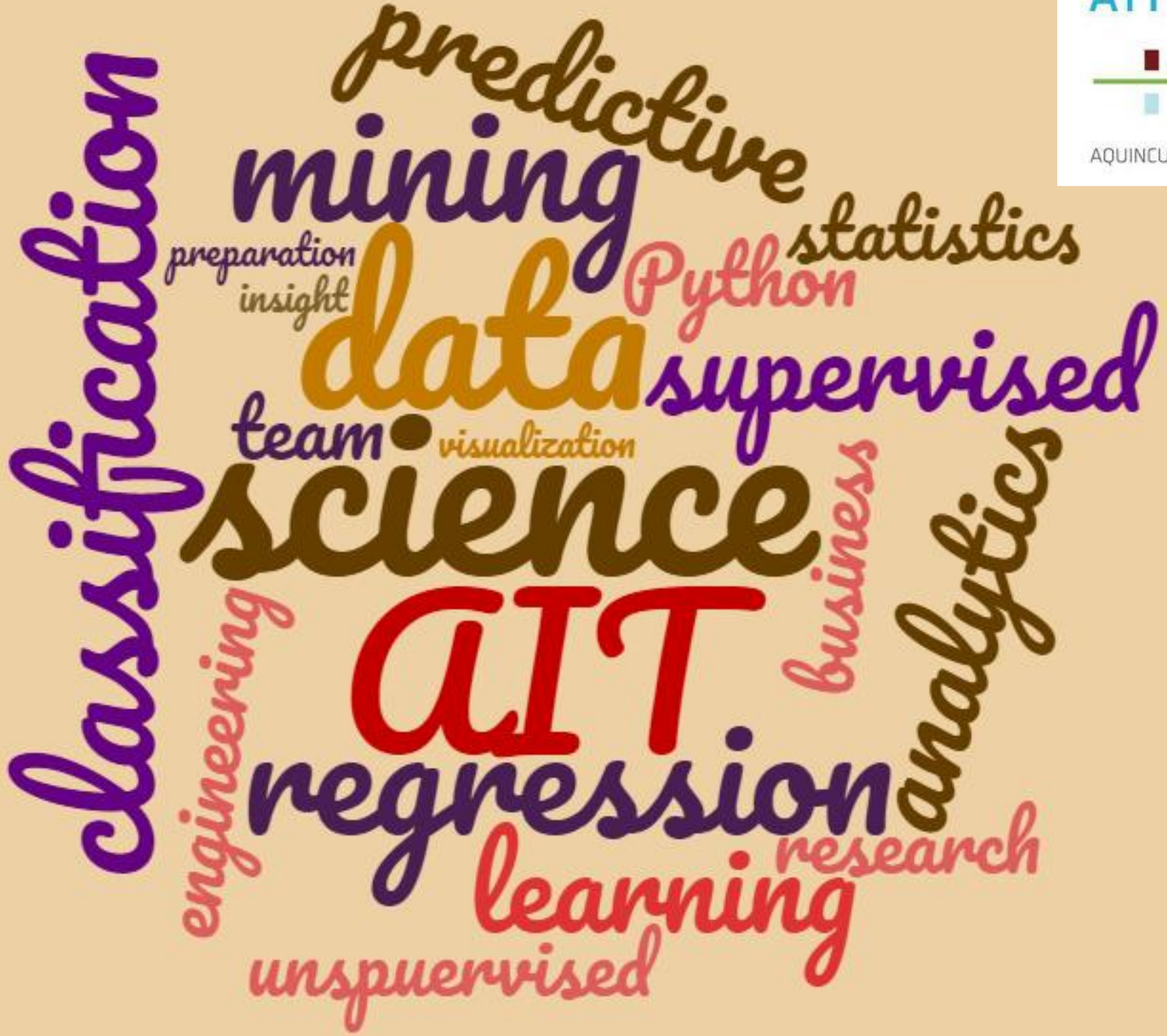
AIT-BUDAPEST

AQUINCUM INSTITUTE OF TECHNOLOGY

Data Science

Roland Molontay

February 11, 2020,
Data preprocessing



Data preparation

- In order to analyze the data, build models, extract knowledge, we need nice, clean structured data
- In most cases we can’t analyze the data in its original form ➔ preprocessing/preparation is necessary
  - Data wrangling, data munging, data cleaning
- It requires a lot of time/work: difficult and time-consuming
- There are no ultimate solutions
- There are some protocols (best practice) that we can follow

Collecting data

- We can use existing datasets gathered by someone else (freely available/commercial)
- Usually not „cut and dried” for our usage
- Sometimes the data that we need can be found in more datasets
  - We have to merge/joint them
- Sometimes the sources of the data are different
  - Possible inconsistencies in the data
- The sources should be documented: where we downloaded from, who is the data owner, who collected the data, what preprocessing steps were conducted
- If somebody has already preprocessed the data, it is still worth understanding what happened during the data preparation
  - Perhaps those features of the data were lost that we are most interested in or it became biased

Main steps of data preprocessing

- Getting to know the data
  - What is the source? How was it collected? Is it still current/up-to-date?
  - What are the attributes? What do they mean?
  - What is the type of each attribute? What are thy typical values?
- Exploratory analysis
  - Creating graphs, diagrams, figures, correlation profiles
- Data cleaning
  - Handling data with bad quality, inconsistent data, missing data
- Selecting important features, dimension reduction

Exploratory analysis

- Goal: make decisions based on the exploratory analysis
  - What algorithms/models to use?
  - What are the important features? Are there any interesting relations or redundancy?
  - Are there any apparent problems with the data that need action?
    - Scaling, missing data, outliers
  - Are there patterns that can be recognized using data visualization?
- Methods:
  - Summary statistics / descriptive statistics
  - Simple plots

Summary statistics

- Purpose: to summarize the variables with numerical values
  - Easy to compute and informative
  - What are the typical values, how scattered are they, what are the frequencies?
  - They can be queried by a simple command in any programs
- Categorical variables: frequencies
- Numerical variables:
  - Percentiles: indicating the value below which a given percentage of observations in a group of observations (e.g. values in a column) falls, e.g. *p*-percentile denotes the x_p value below which *p*% of the observations may be found
    - Usually considered values: min, 25, 50, 75, max
  - Mean: arithmetical average of the values: $mean(x) = \bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$
    - Sensitive to outliers median is more robust
  - Median: the value separating the higher half from the lower half of a data sample
    - Similar to the 50-percentile but not the same

$$median(x) = \begin{cases} x_{r+1} & \text{if } m = 2r+1 \\ \frac{1}{2}(x_r + x_{r+1}) & \text{if } m = 2r \end{cases}$$

Values describing deviation

- Range: what is the range of the possible values: *max - min*
- Sample variance:
  - Sensitive to outliers

$$s_x^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \overline{x})^2$$

  - Standard deviation: root of the variance
- Average absolute deviation:

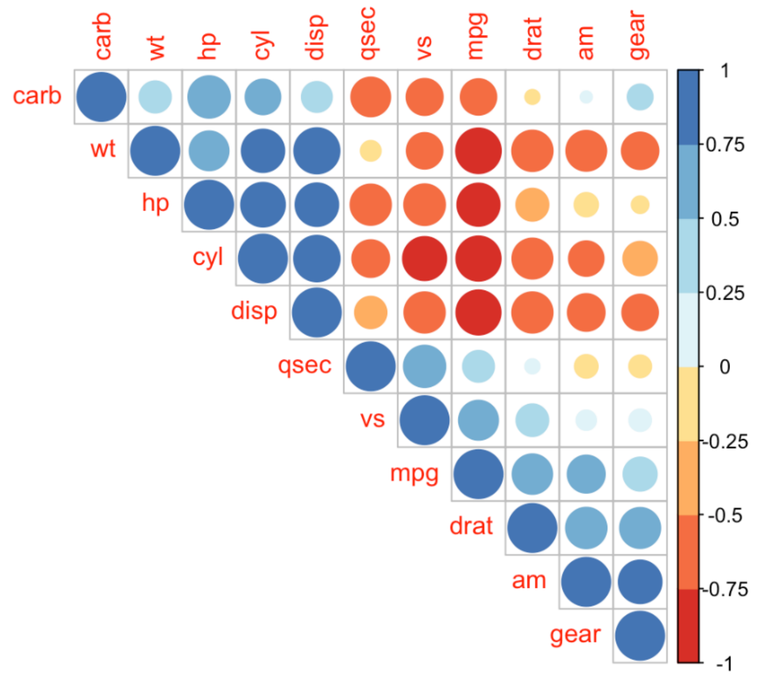
$$\frac{1}{m}\sum_{i=1}^{m}|x_i - \overline{x}|$$

Covariance and correlation

- Sample covariance
  between values of *j*th and *k*th columns (attributes)

$$q_{jk} = \frac{1}{m-1}\sum_{i=1}^{m}(x_{ij}-\bar{x}_j)(x_{ik}-\bar{x}_k)$$

  - We can form a matrix: sample covariance matrix (symmetric)
- Sample correlation: $r_{jk} = \frac{q_{jk}}{s_j s_k}$
  - A sample correlation matrix can be formed
- Not robust against outliers
- Measure the strength of the linear relationship between variables

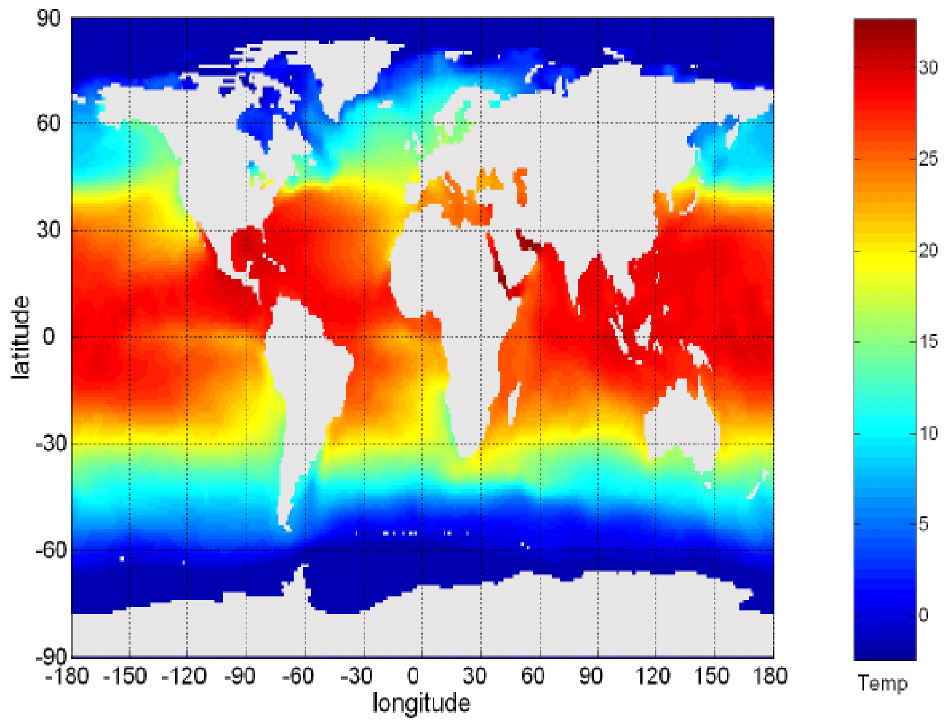


Visualization

- In the exploration stage the aim is to explore the relationship between variables and to display the characteristics of the data in a way that is easy to perceive by a human
  - It is easier to interpret a graph than a table
  - The patterns are more apparent
  - Outliers, anomalies could be recognized
- In this stage we create many quick figures, the aim is not to have fancy, sophisticated figures (later, in the presentation stage)

Good visualization

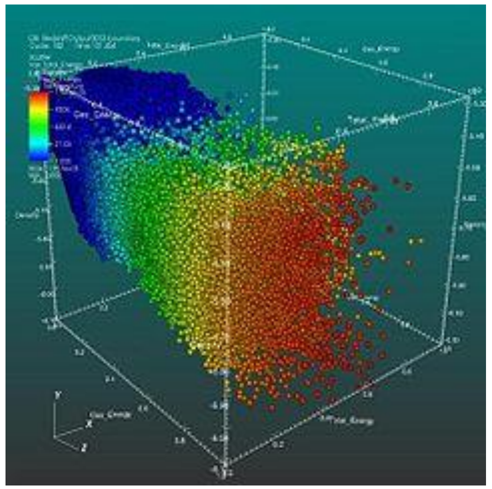
- The layout and interpretability is important
  - In the exploration stage it is sufficient if we understand the figure
  - In the explanation/presentation stage the audience must also understand it

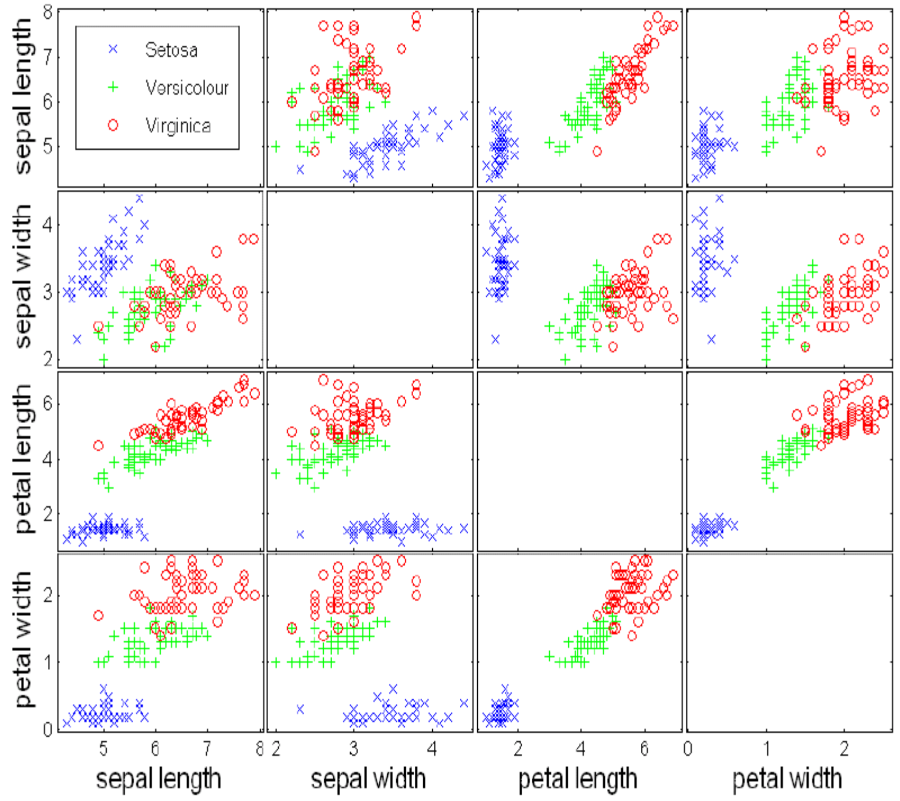


Sea surface temperature
Thousands of data points visualized in on figure

Scatterplot

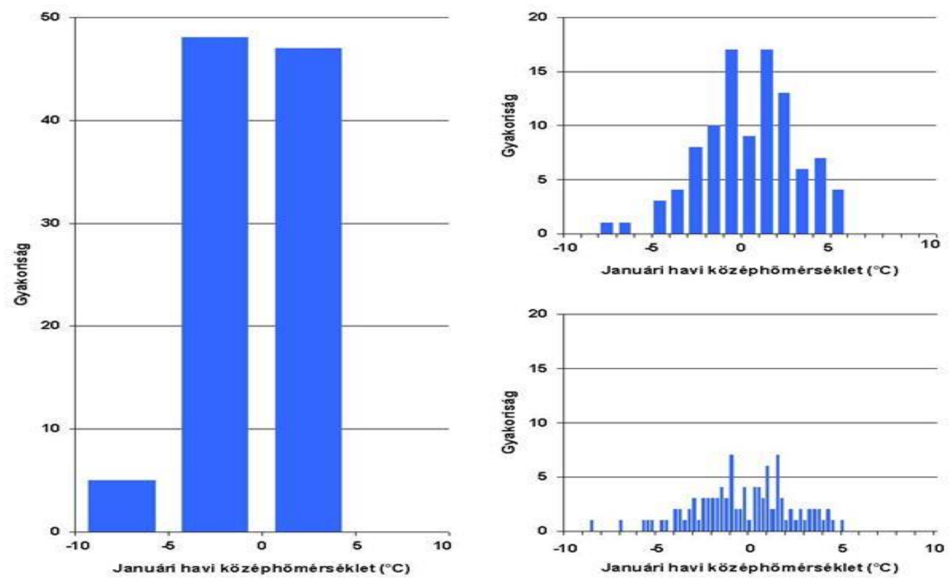
- The objects correspond to points on the plane / in the space
- The coordinates of the points correspond to the values of two/three attributes of the object
- Beyond the (max) three dimensions the point can also have color/shape/size
    - So we can visualize 5-6 dimension all together
        - It is hard to interpret above 4 dimensions

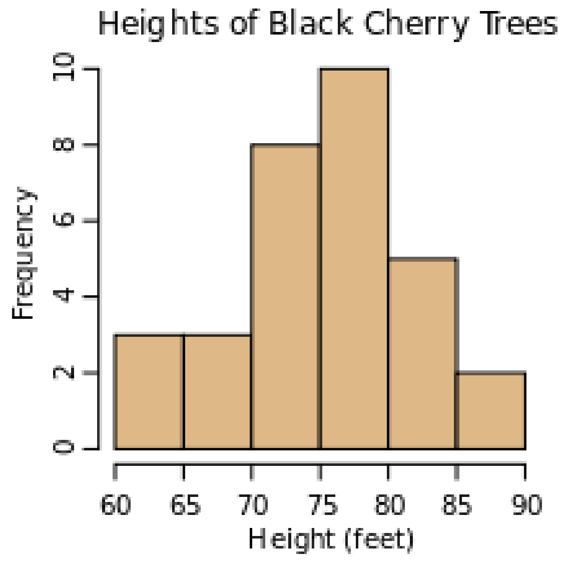




Histogram

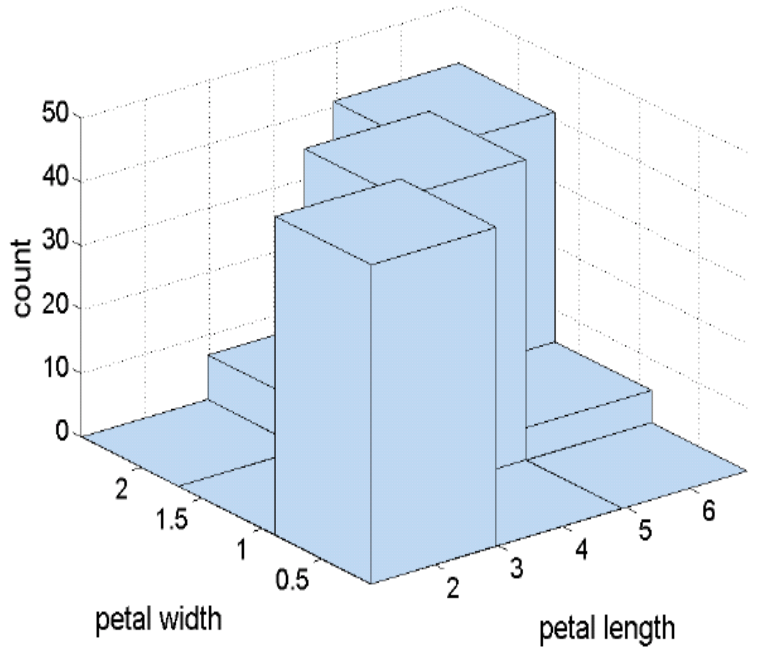
- Representation of the distribution of numerical data
- It is an estimate of the probability distribution of a continuous variable
  - Empirical distribution
- Binning the range of values: dividing the entire range of values into a series of intervals
- Counting how many values fall into each interval
  - A rectangle is erected over the bin with height proportional to the frequency

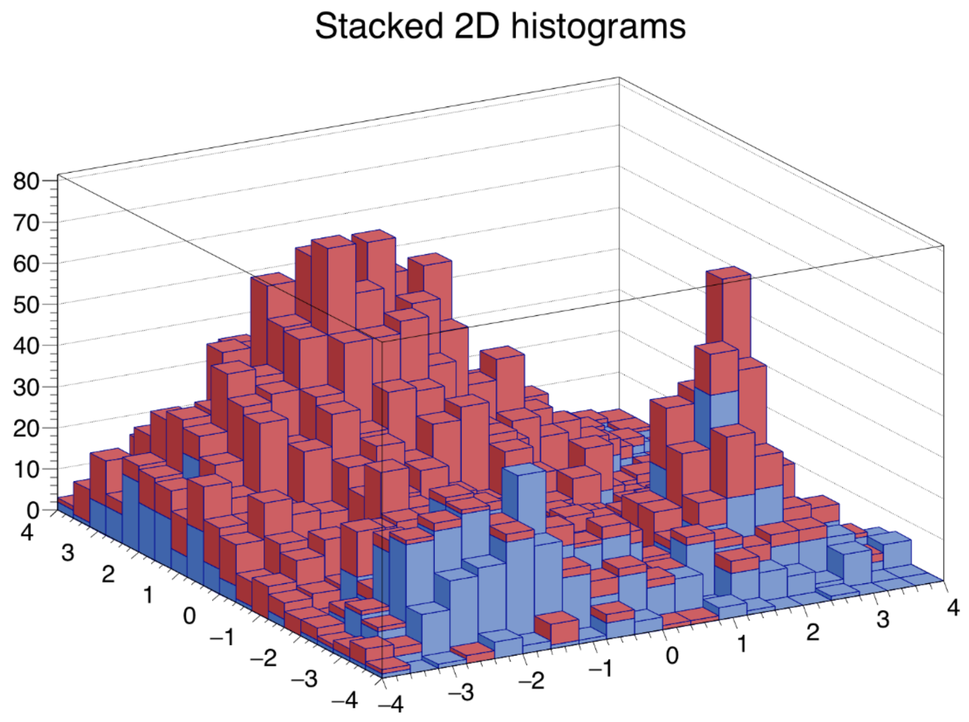




Two-dimensional histogram

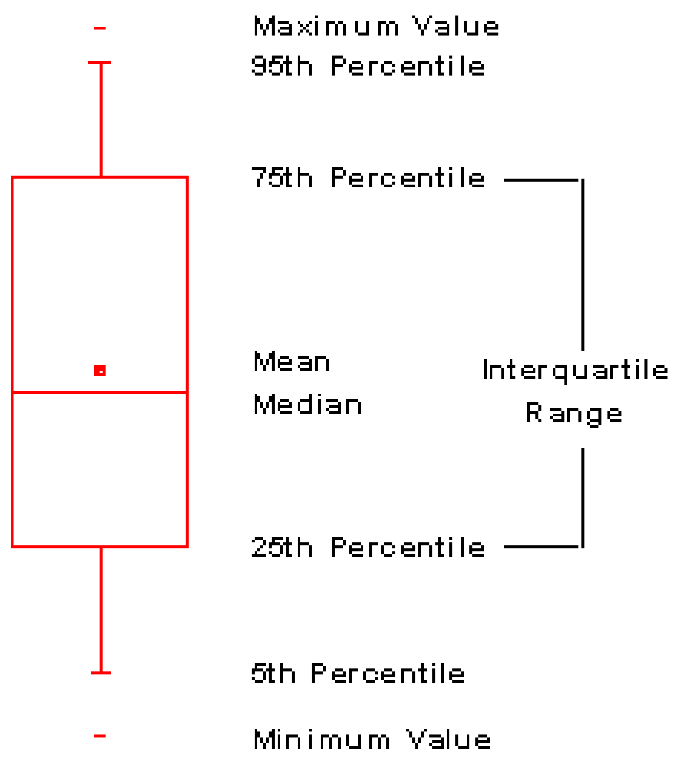
- It estimates the joint distribution of two variables

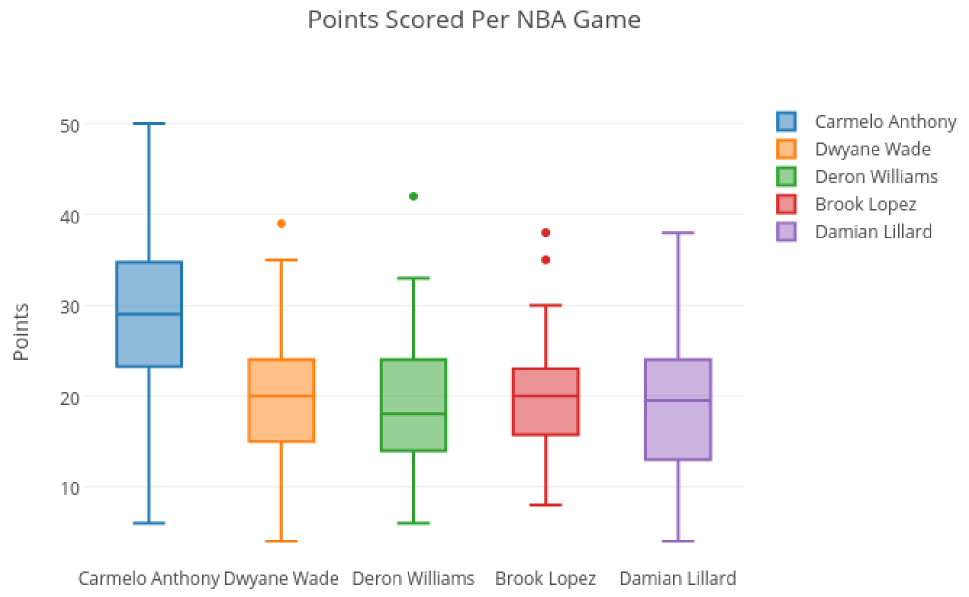




Boxplot

- Another method to visualize distribution
  - Attributed to J. Tukey





Reducing the number of attributes

Aim: to have fewer columns

Why?

- Achieve faster running time
- Need less storage capacity
- Easier to visualize
- The results are easier to interpret
- In high dimension most of the models perform poorly (due to curse of dimensionality)

How?

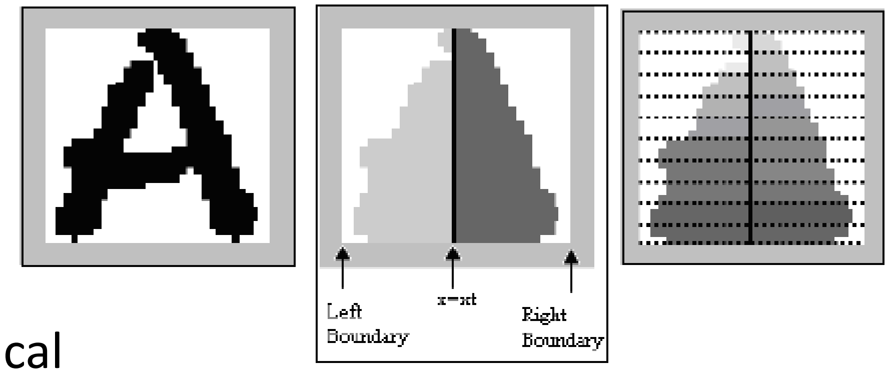
- Omit redundant columns
- Merging columns
- Introducing new (better) attributes and omitting the old ones
- Dimension reduction methods (later)

Methods

- Finding redundant columns
  - E.g. Column A: price of the product, Column B: paid VAT
- Finding irrelevant columns
  - E.g. the phone number of a person regarding his/her creditworthiness (are you sure?)
- Brute force: checking all possible subsets of the attribute set (be careful: 2^d)
- Automatic filtering
  - If the correlation of two columns is too high, we omit one of them
- Embedded methods
  - The used machine learning method chooses the relevant variables itself (later)
- Other dimension reduction methods (e.g. PCA)
- After reducing the number of attributes we can evaluate whether the result/running time of the algorithm improved

Introducing new attributes

- Sometimes we don’t necessarily want to reduce the number of attributes but we want better, more expressive attributes
- Sometimes domain knowledge is needed
- Examples:
  - To extract features from pixel series of images
    - Number of „on” pixels, average of the horizontal coordinates of the „on” pixels, variance of the vertical coordinates, correlation between the horizontal and vertical positions of „on” pixels
  - Combining attributes based on domain knowledge
    - Introducing density instead of volume and mass



Transforming attributes

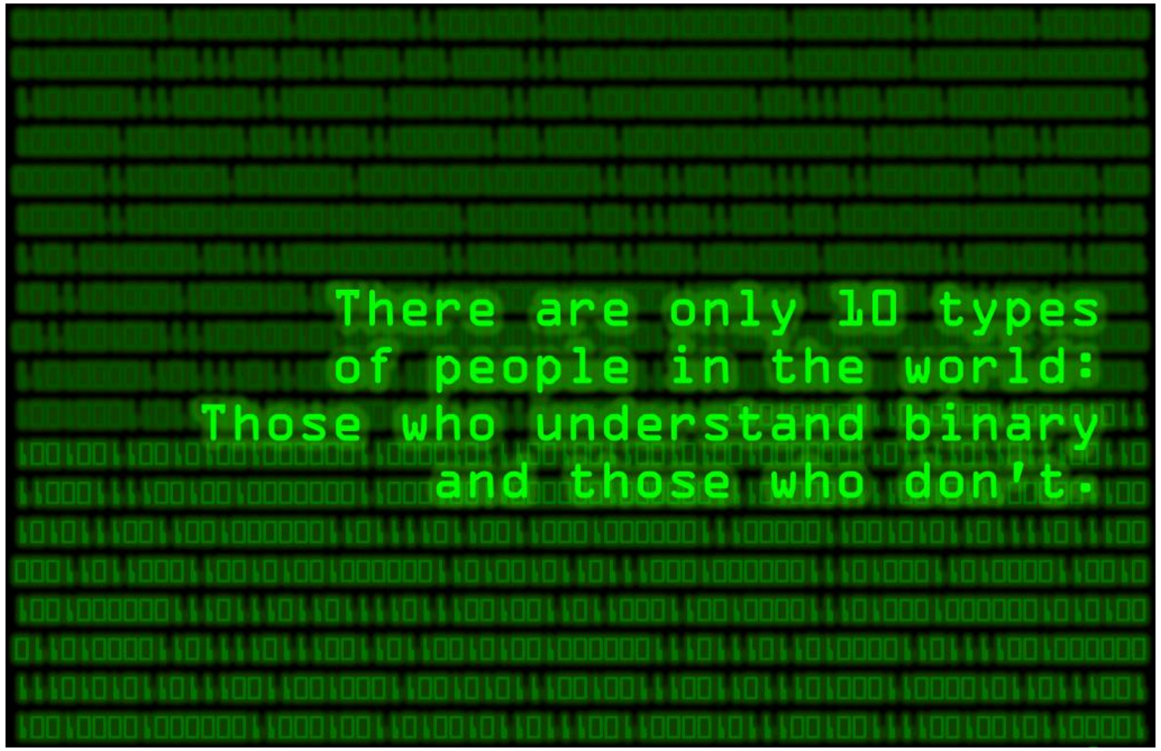
- Feature scaling: sometimes it is necessary to standardize/normalize the range of independent variables (e.g. for visualization, for some machine learning algorithms)
  - Rescaling the range in [0, 1] (min-max normalization) : $$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$
  - Standardization (zero-mean, unit-variance): $$x' = \frac{x - \bar{x}}{s_x}$$
  - Scaling to unit length: $$x' = \frac{x}{||x||}$$
- Bijective mapping of an attribute
  - E.g. in some application we can take the logarithm of the salary (it makes more sense to measure „percent” changes in wage rather than absolute changes and it is usually more normally distributed)

Discretization

- Goal: transferring continuous variables into discrete (nominal) counterparts
- Why?
    - Some algorithms need discrete variables (e.g. association rules)
    - We need to sore less values, for some algorithms the running time is much faster
    - Sometimes a rougher scale is sufficient, e.g. high, medium, low values, the data could be more clear-out
- How to partition the data? What are the discretization cut points?
    - Divide the range of the continuous variable into intervals of the same length (equidistant division)
    - Dividing the range into intervals with the same number of observations (along quantile values) – equal frequency
    - Dividing along quantile values creating groups with differing size, e.g. along quantiles 10, 30, 70, 90
    - If there are some natural cut points where the data is rare, it is reasonable to choose these cut points

From categorial to binary

- If a nominal attribute has *k* possible values, it is replaced by *k* synthetic binary attributes (one attribute-per-value approach or one-hot encoding)
  - The *i*th being 1 if and only if the original value corresponds to the *i*th group
- Another approach using *k-1* synthetic binary attributes: the *i*th being 0 if the value is one of the first *i* in the ordering and 1 otherwise



Acknowledgement

- András Benczúr, Róbert Pálovics, SZTAKI-AIT, DM1-2
- Krisztián Buza, MTA-BME, VISZJV68
- Bálint Daróczy, SZTAKI-BME, VISZAMA01
- Judit Csima, BME, VISZM185
- Gábor Horváth, Péter Antal, BME, VIMMD294, VIMIA313
- Lukács András, ELTE, MM1C1AB6E
- Tim Kraska, Brown University, CS195
- Dan Potter, Carsten Binnig, Eli Upfal, Brown University, CS1951A
- Erik Sudderth, Brown University, CS142
- Joe Blitzstein, Hanspeter Pfister, Verena Kaynig-Fittkau, Harvard University, CS109
- Rajan Patel, Stanford University, STAT202
- Andrew Ng, John Duchi, Stanford University, CS229