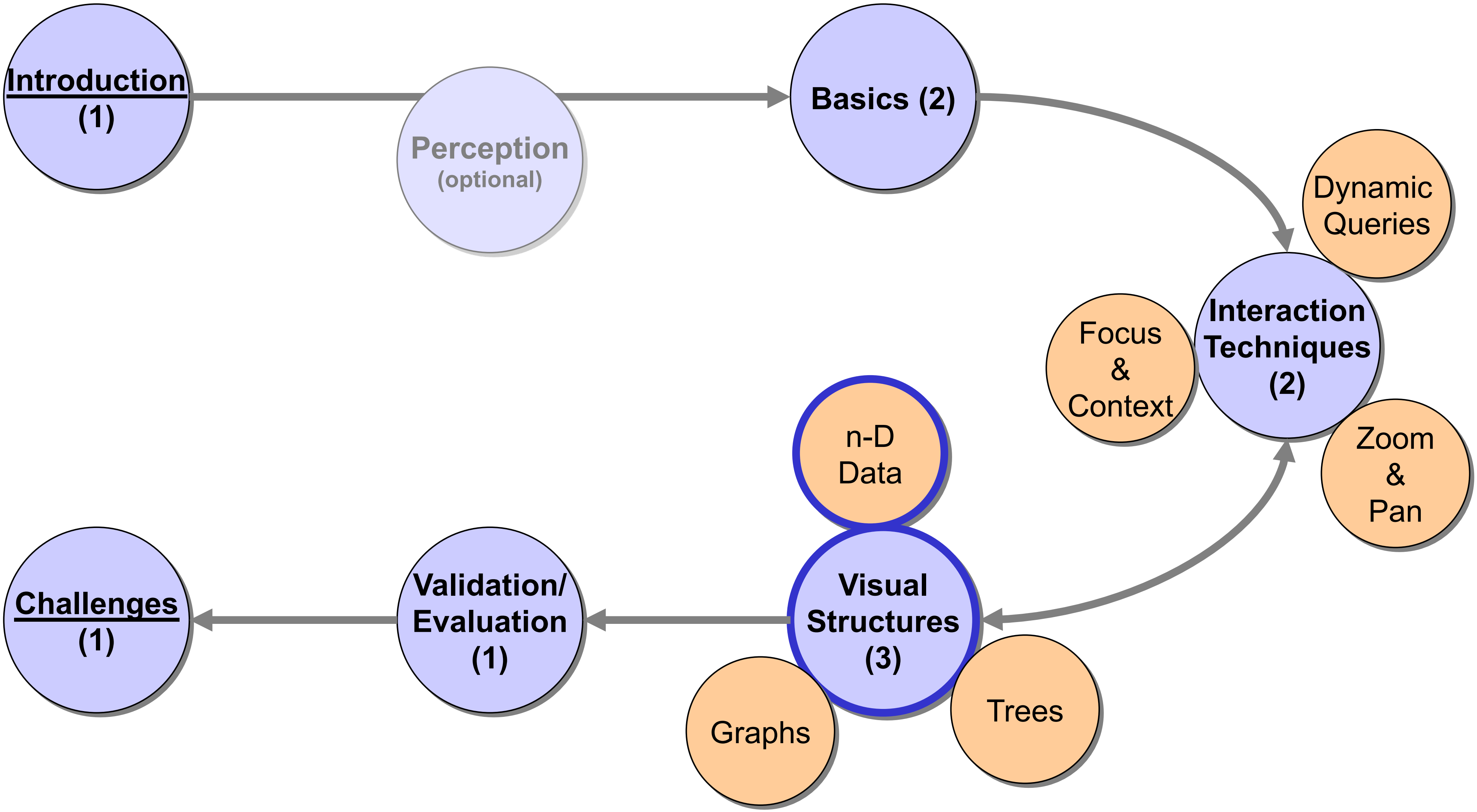
Information Visualization

4. 1D, 2D, 3D, and Multidimensional Data

TNM111: 10 Lectures



4.1 Repetition

- Data Tables
  - Cases/Items/Objects ☐
  - Variables (Attributes) ☐
    - Nominal
    - Ordinal
    - Quantitative
  - Values ☐
  - Metadata (e.g., V-Type) ☐

a possible data table

Model	V-Type	120d	2.0 TDI
Manufac.	N	BMW	AUDI
Series	N	1er	A4
ID	O	36983	43832
HP	Q	163	140

4.1 Repetition

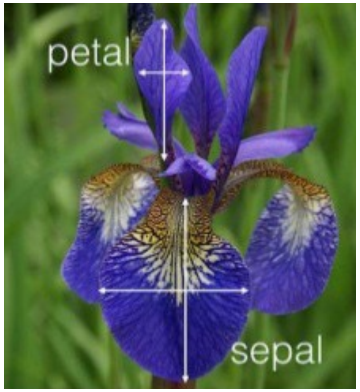
- The mapping to visual structures depends on the dimensionality of the data
  - Depends of the number of attributes (variables)
  - 1D → univariate data
  - 2D → bivariate data
  - 3D → trivariate data
  - ≥4D → multivariate data (*hypervariate* data)

[Spe:34ff]

4.1 Repetition

- Example of multidimensional data (*Iris data set*)

Sepal length	Sepal width	Petal length	Petal width	Species
5.2	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
4.6	3.1	1.5	0.2	*I. setosa*
5.0	3.6	1.4	0.3	*I. setosa*
5.4	3.9	1.7	0.4	*I. setosa*
4.6	3.4	1.4	0.3	*I. setosa*
5.0	3.4	1.5	0.2	*I. setosa*
4.4	2.9	1.4	0.2	*I. setosa*
4.9	3.1	1.5	0.1	*I. setosa*
5.4	3.7	1.5	0.2	*I. setosa*
4.8	3.4	1.6	0.2	*I. setosa*
4.8	3.0	1.4	0.1	*I. setosa*
4.3	3.0	1.1	0.1	*I. setosa*
5.8	4.0	1.2	0.2	*I. setosa*
5.7	4.4	1.5	0.4	*I. setosa*
5.4	3.9	1.3	0.4	*I. setosa*



3 species

4.1 Repetition

- Further data types
  - Trees
  - Networks (graphs)
  - Media data [→ TNM098 (partly)]
    - Text documents, images, sound data, movies, …
    - Multimedia (linked media data)
  - Processes [→ TNM098 (partly)]
    - Time-dependent data
    - Algorithms and programs
    - Evolutionary processes
  - …

4.2 Introduction

- In InfoVis, we try to map all data to the 2D or 3D space (visual mapping)
  - Uni-, bi- and trivariate data (in the following mostly quantitative)
    - Visualizations often code information through positioning of marks on orthogonal axes
  - Multivariate data (in the following mostly quantitative)
    - Orthogonal visual structures cannot help here
    - Multiple axes, pixel, icons, …
- Again, visual perception is very important for visual mapping and the design of visual structures!

4.3 Univariate Data

- Basis is a sequence of one-dimensional, numeric values
- Classic idea
  - Enter the values along *one* axis
  - Represent the values or objects as
    - dots
    - bars
    - segments of circles
    - etc.

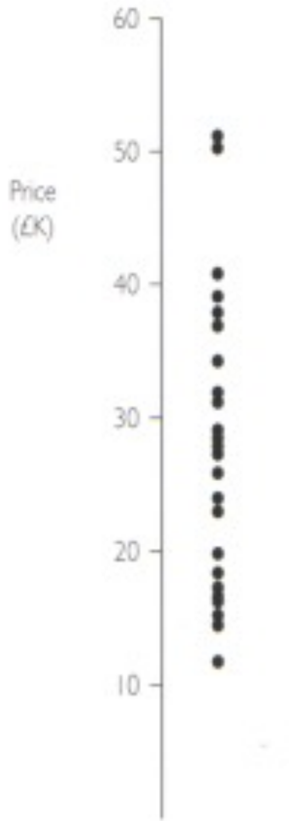
  → 1D, 2D, 3D visual structures

The price and make of a collection of cars

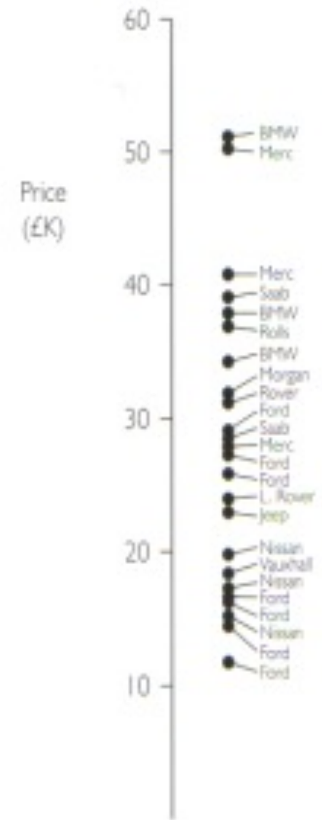
Car	Price (£)
BMW	51,395
Mercedes	50,850
Mercedes	41,000
Saab	39,085
BMW	38,000
Rolls	36,950
BMW	34,550
Morgan	32,000
Rover	31,300
Ford	29,250
Saab	28,750
Mercedes	28,000
Ford	27,600
Ford	25,950
Land Rover	24,000
Jeep	23,200
Nissan	20,000
Vauxhall	18,500
Nissan	17,400
Ford	17,000
Ford	16,500
Nissan	15,500
Ford	14,950
Ford	12,000

4.3.1 Representations

- Plots (1D)

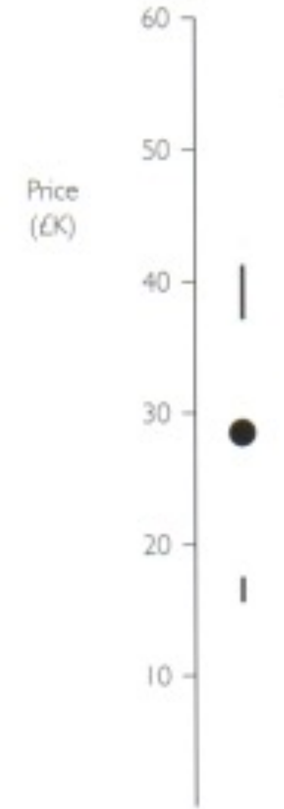


Plot



Plot with labels

Tukey Box Plot

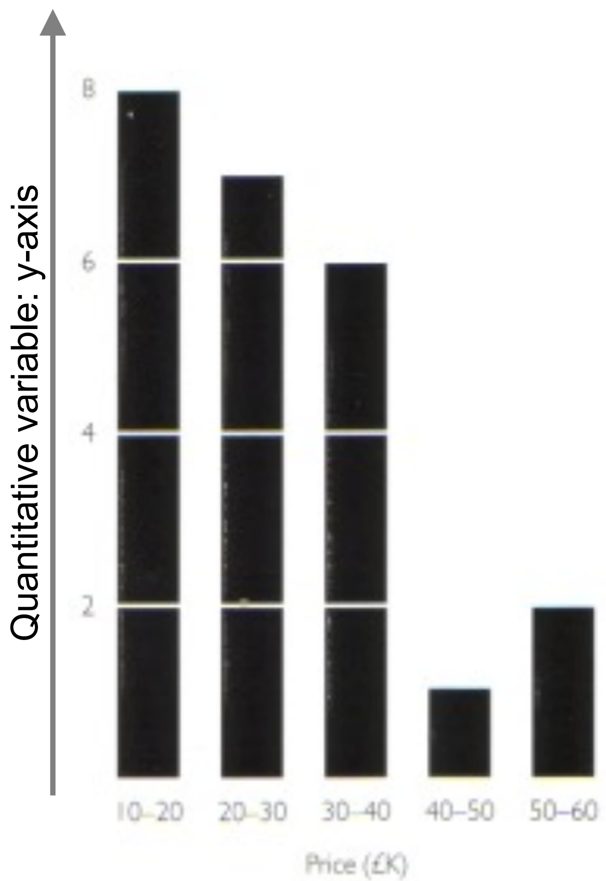


Alternative representation

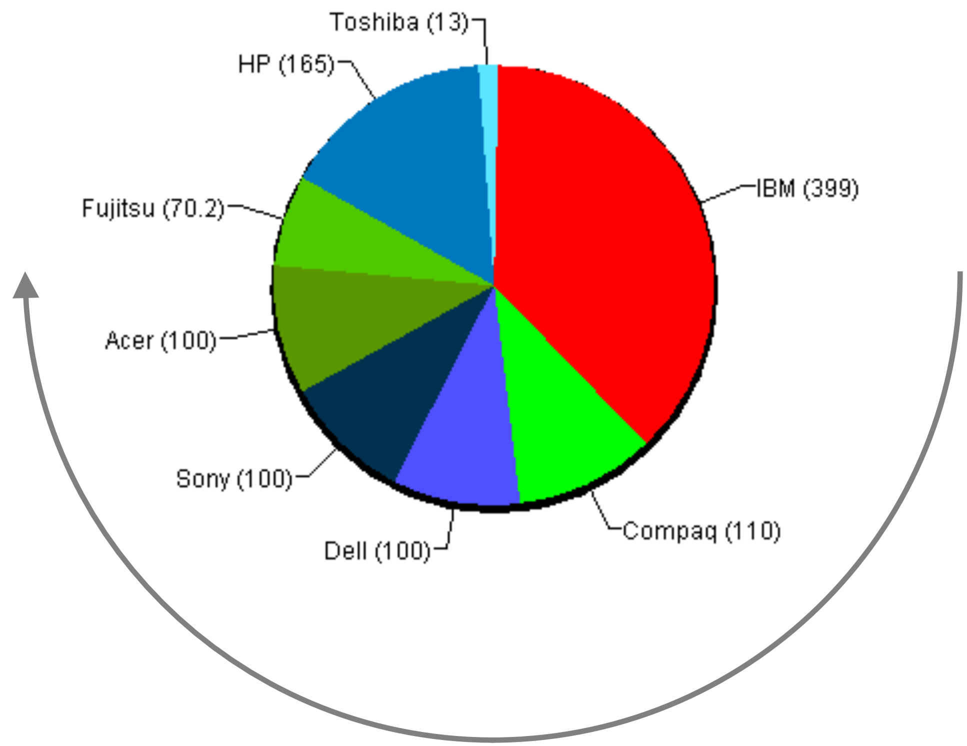
[Spe:36ff]

4.3.1 Representations

- Histograms (2D)



- Pie charts (2D)



Quantitative variable: circle

- Semantic zooming

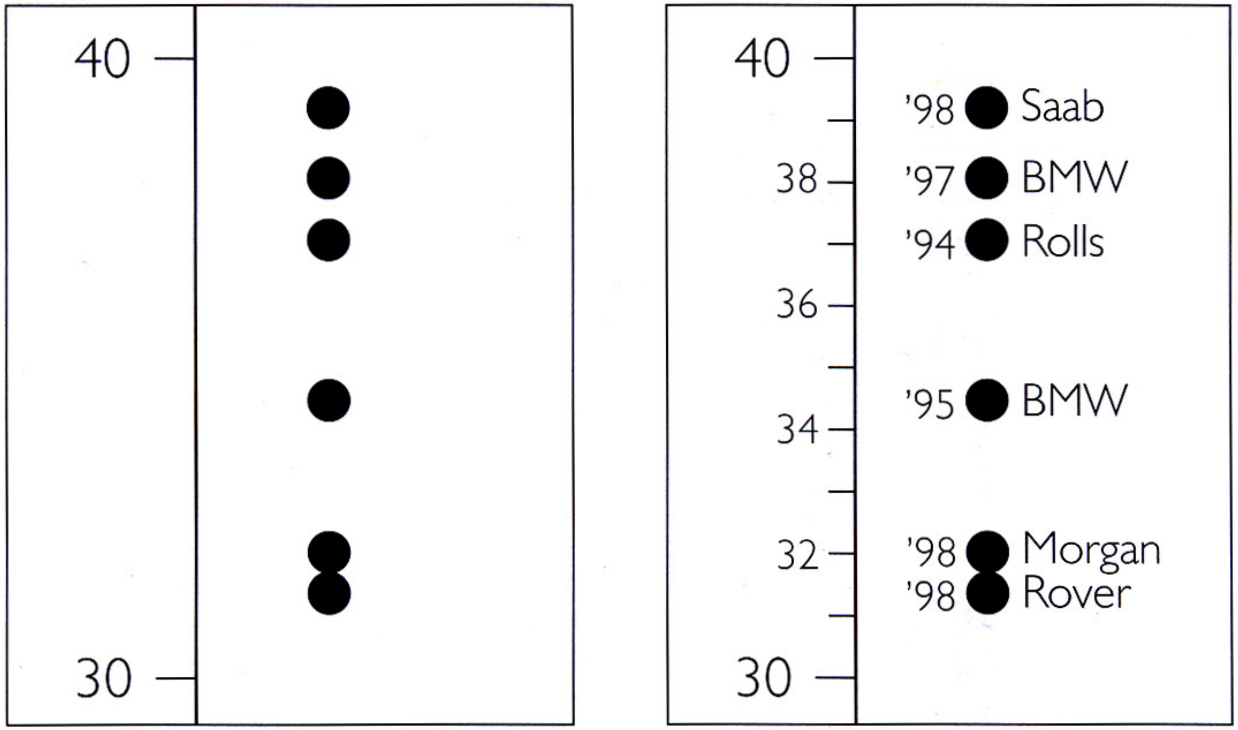


FIGURE 3.7
Result of zooming into univariate data

FIGURE 3.8
Result of a logical zoom

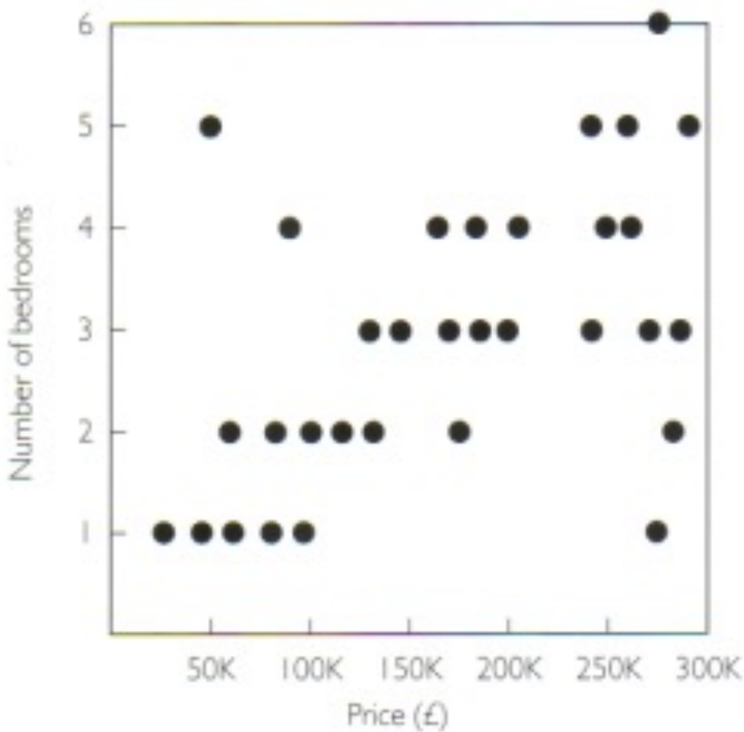
4.4 Bivariate Data

- Basis is a sequence of two-dimensional, numeric values
- Classic idea
  - Enter the values along *two* axes
  - Represent the values as
    - dots or marks

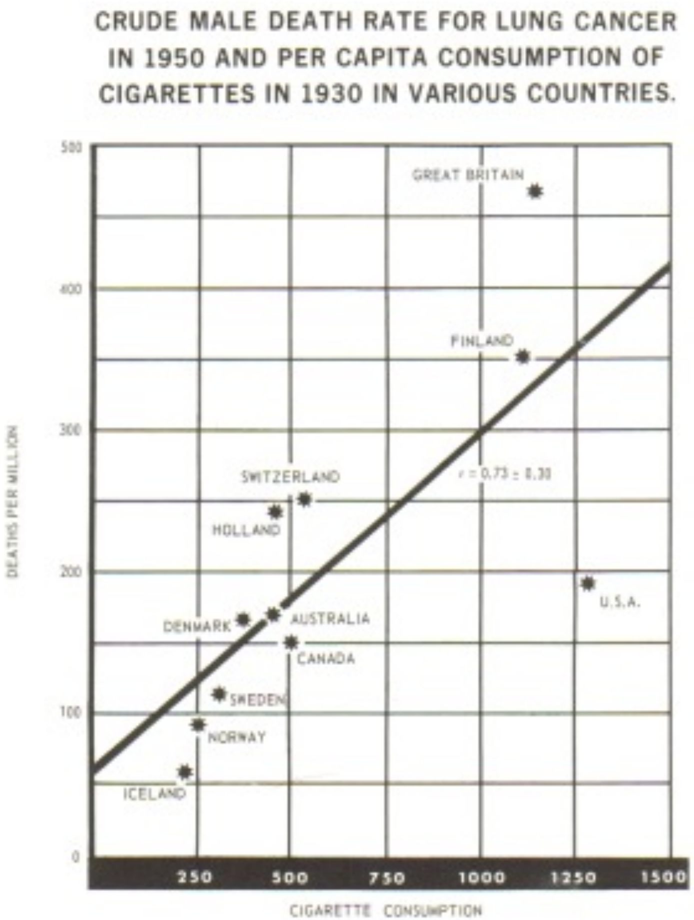
  → 2D, 3D visual structures
- Aim of the visualization of bivariate data
  - Recognize new correlation between two values, e.g., linear or curved patterns, etc.

4.4.1 Representations

- Scatter Plots (2D)



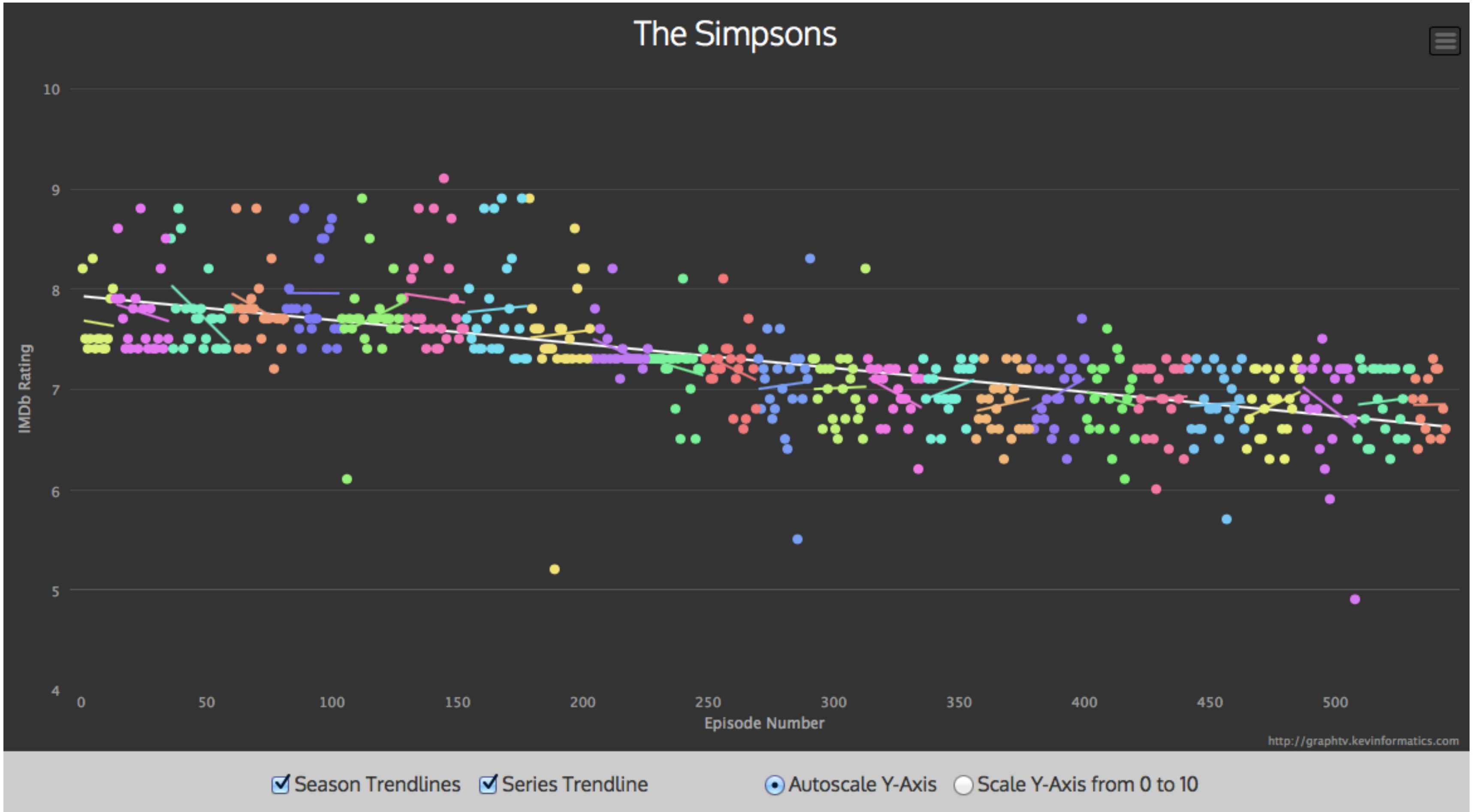
- Scatter Plots with additional statistic information



[Spe:39ff]

4.4.1 Representations

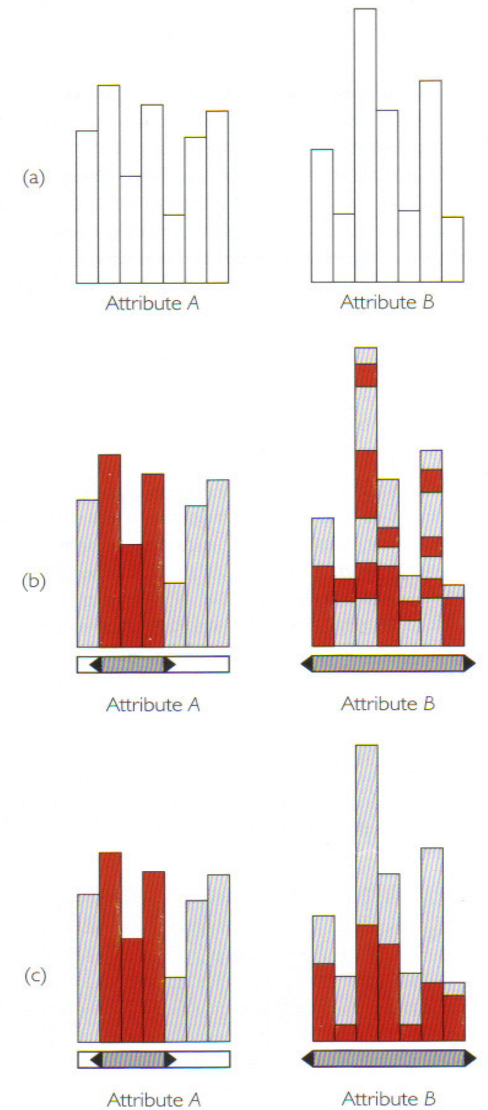
- Online scatterplot example with several regression lines



[taken from the “Graph TV” web site (shut down)]

4.4.1 Representations

- Histograms (2D)
  - Separate histograms
  - Coupled histograms
    - Coupling with the help of
      - color
      - patterns
      - …
  - Interaction
    - Sliders (dynamic queries)



(a) Histograms of two attributes
(b) Histograms with adjustable limits for selection purposes
(c) Selected objects are 'ranged down' for easier interpretation

4.5 Trivariate Data

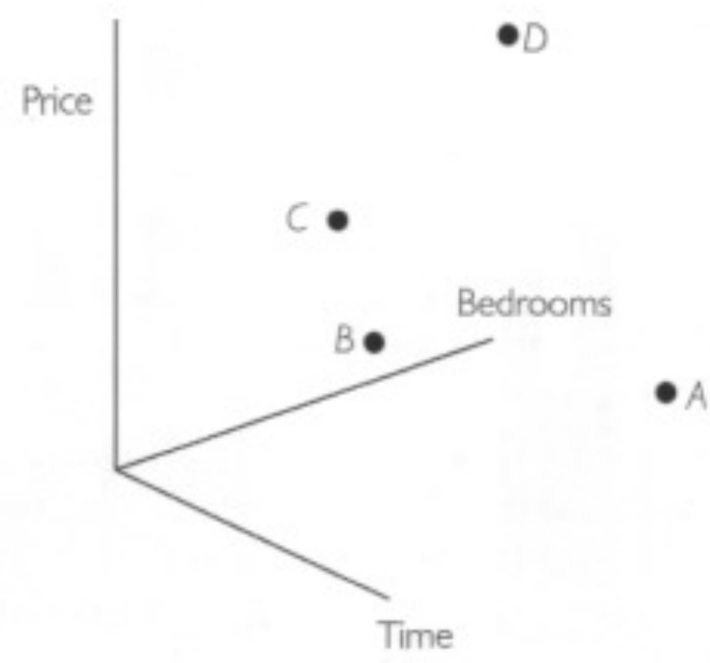
- Basis is a sequence of three-dimensional, numeric values
- Classic idea
  - Enter the values along *three* axes
  - Represent the values as
    - dots, spheres, or marks

    → 3D visual structures (or logic/geometric 2D projections from 3D)
- Aim of the visualization of trivariate data
  - Recognize new correlation between three values, e.g., linear or curved patterns etc.

4.5.1 Representations

- 3D Scatter Plots
  - In this example, the 2D representation seems to lead to a lot of problems if we have many cases

FIGURE 3.12
A 2D presentation of a 3D plot



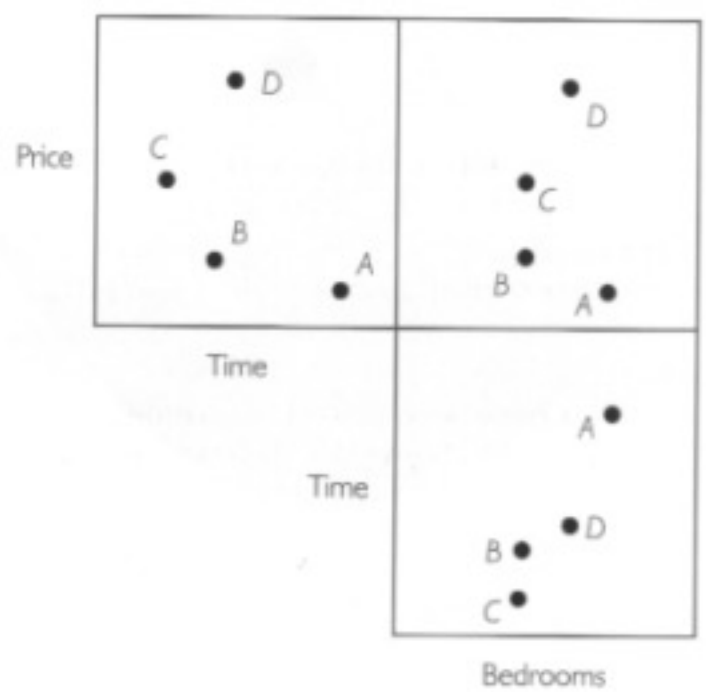
[Spe:42ff]

4.5.1 Representations

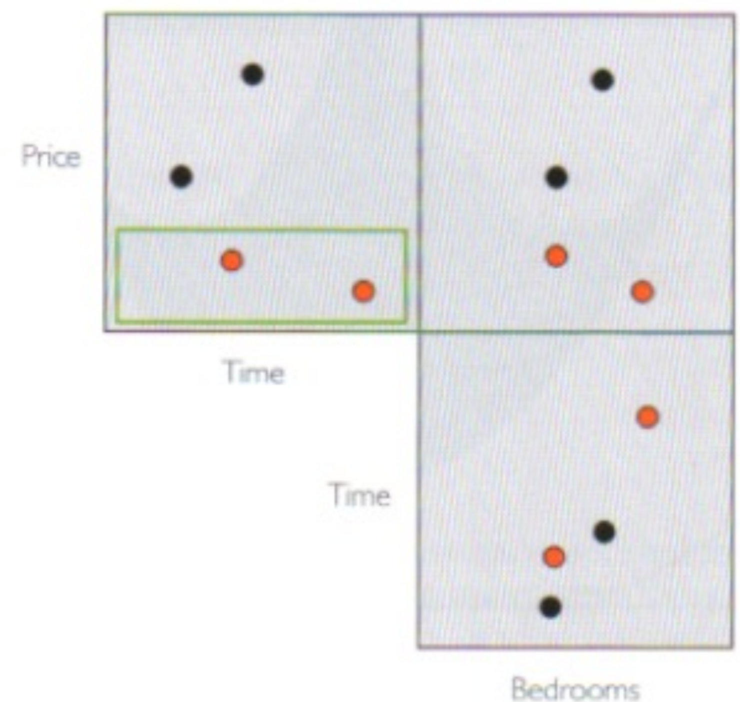
- 3D Scatter Plots (cont.)
  - A solution would be projections along all three axes

4.5.1 Representations

- Scatter Plots Matrices
  - They offer a good overview (also for higher dimensions, i.e., for multivariate data)



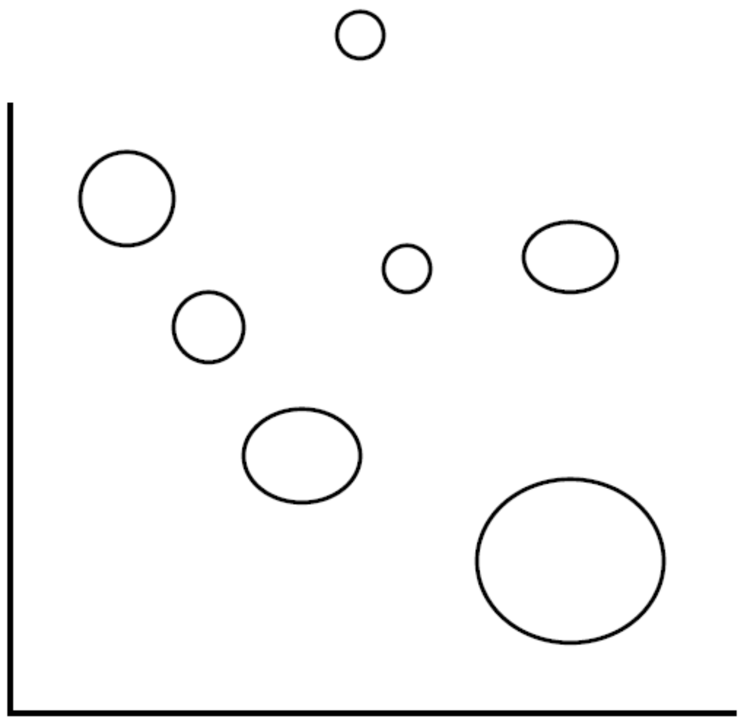
Scatter Plot Matrix



Scatter Plot Matrix with brushing

4.5.1 Representations

- The examples presented before show that we can display each variable separately
  - This is always possible, i.e., for multivariate data too!
- Another solution is to display one dimension with another coding scheme, for example, size, shape, or color:

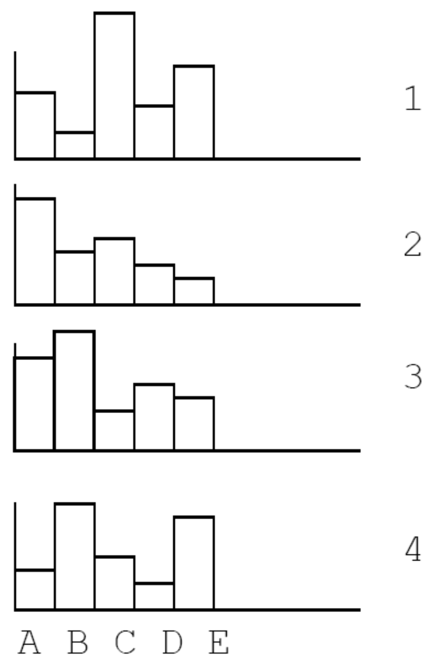


[Inspired by J. Stasko's course]

4.6 Multivariate Data

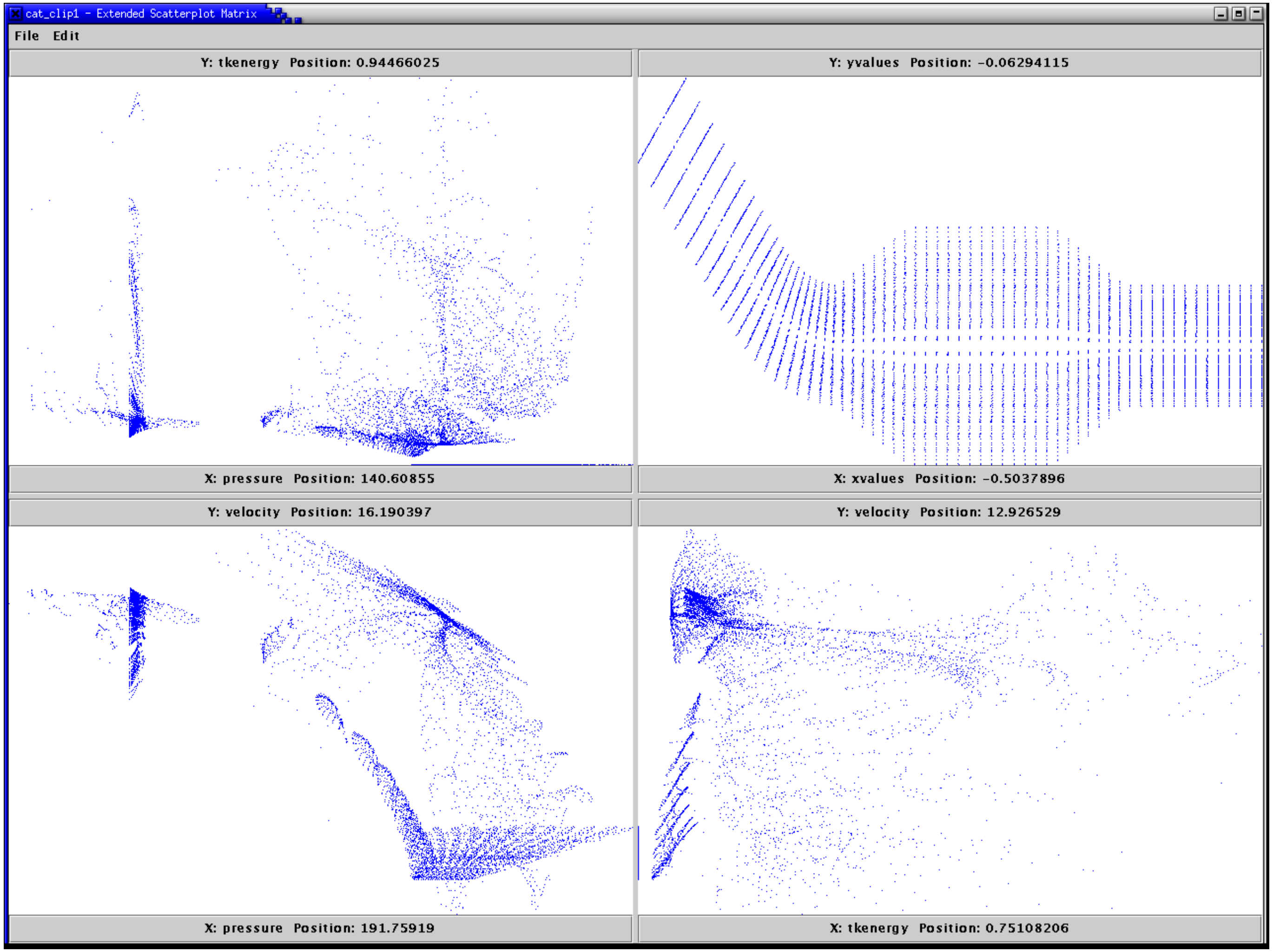
- Multivariate Data demand for non-traditional solutions [now, it will become interesting]
- We have seen two possible solutions, but they complicate to see complex correlations:
  - Usage of one separate view for each variable/dimension

	A	B	C	D	E
1	4	1	8	3	5
2	6	3	4	2	1
3	5	7	2	4	3
4	2	6	3	1	5



4.6 Multivariate Data

- Scatter Plot Matrices

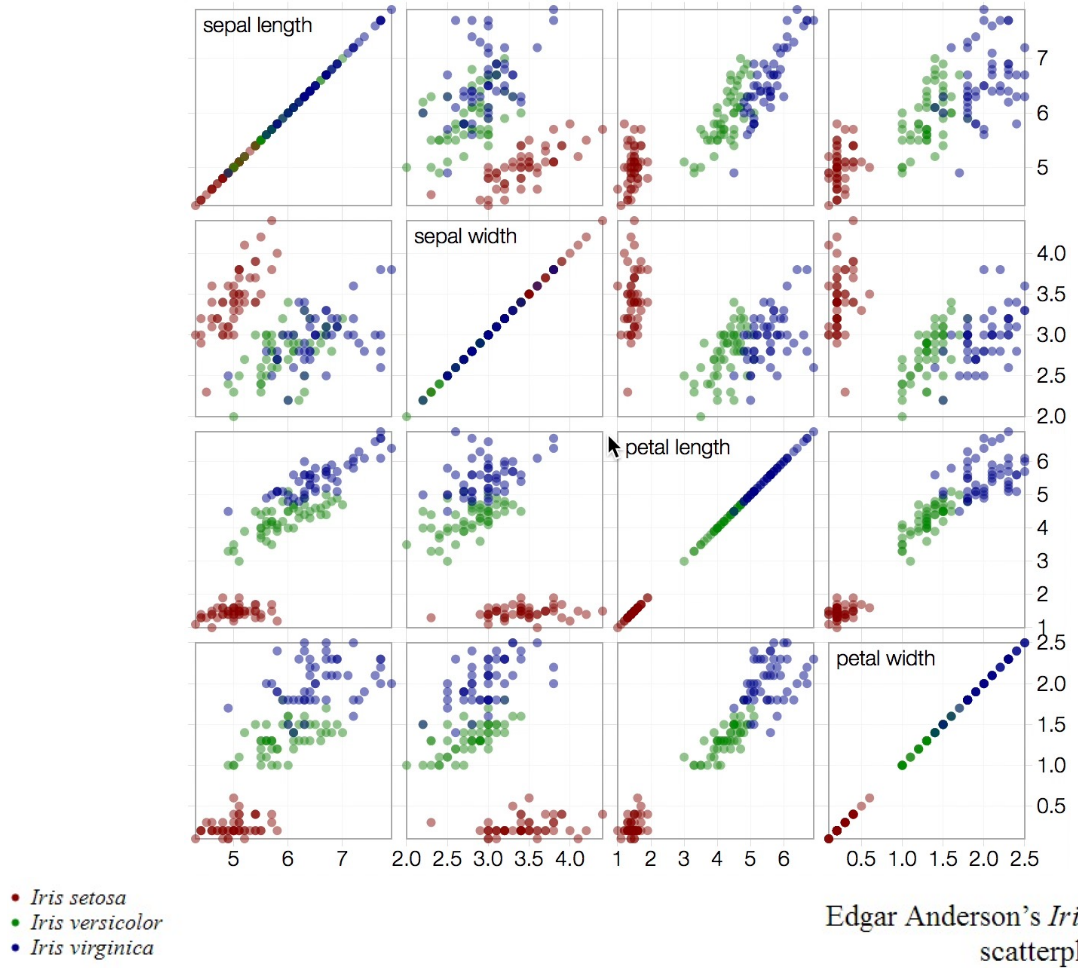


http://www.vrvis.at

4.6 Multivariate Data

- There are many different visualization techniques to sufficiently display multivariate data
  - See http://sci.utah.edu/~shusenl/highDimSurvey/website/ for a survey
- We know four different types
  - Projection-based approaches
    - Scatter Plot Matrices [already briefly presented], …
  - Coordinate axis-based approaches
    - Parallel Coordinates, Star Plots, …
  - Icon-based approaches
    - Chernoff Faces, Stick Figures, …
  - Pixel-based approaches
    - Recursive Patterns, SeeSoft, …

- Scatterplot Matrix (in action)



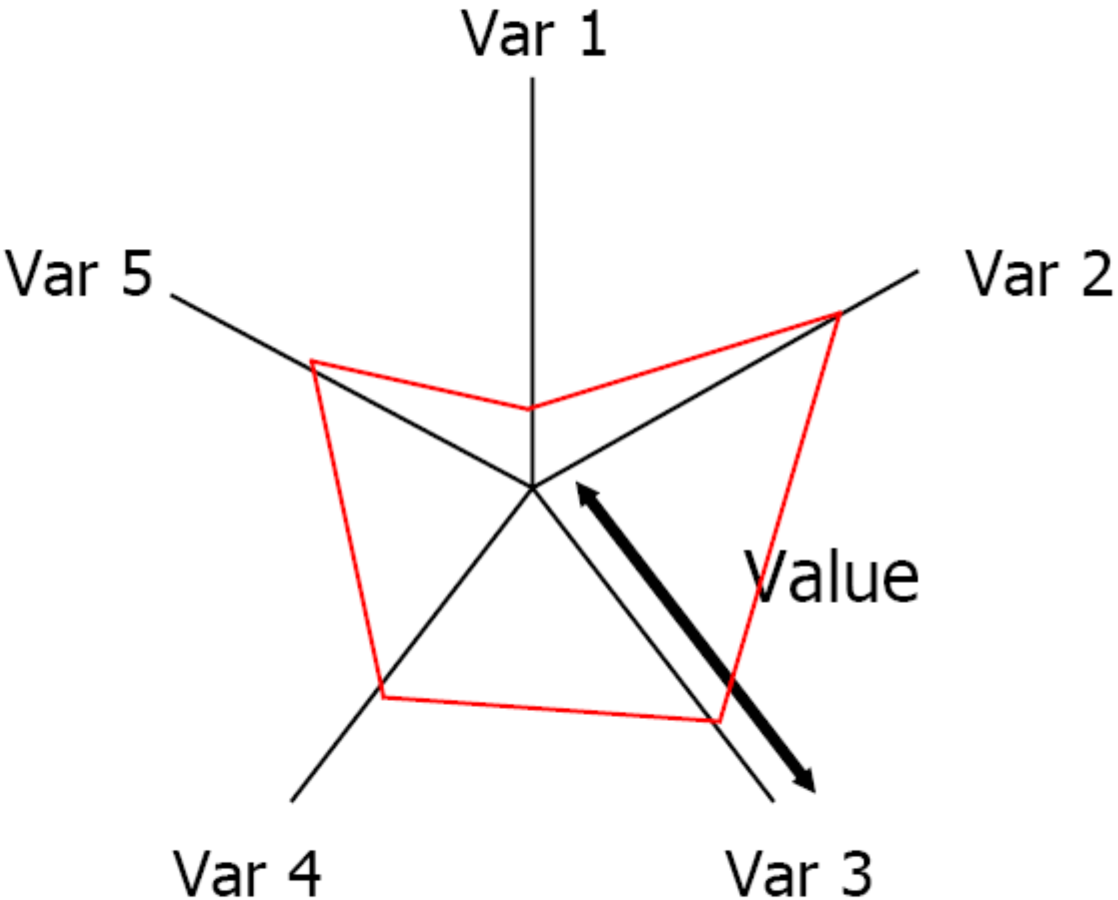
[Demo]

Edgar Anderson's *Iris* data set scatterplot matrix

4.6.2 Axis-based Approaches

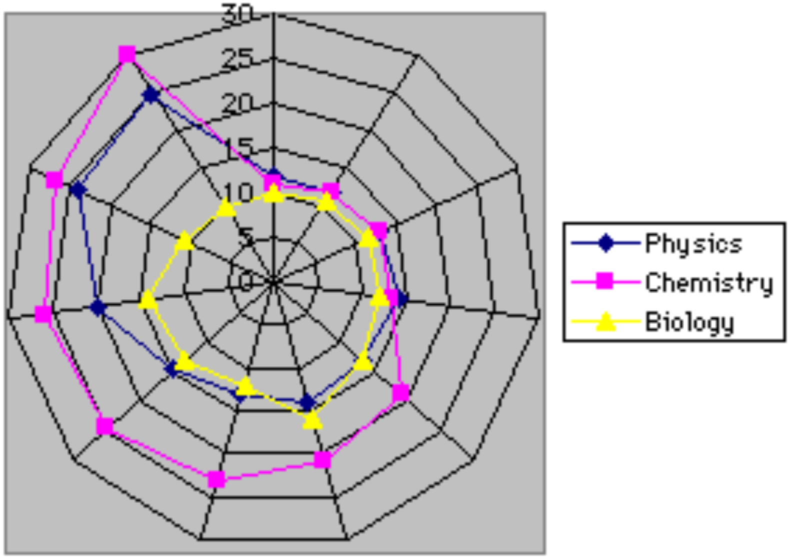
Star Plots

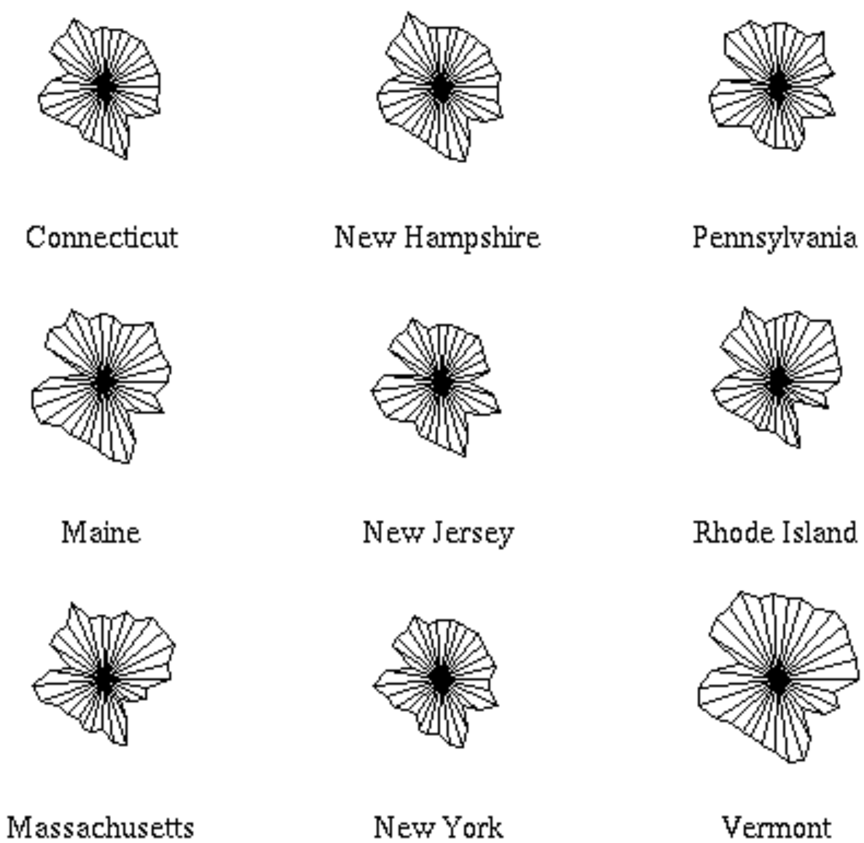
- Firstly, draw the axes like a star
- Each „spoke“ codes a variable value
- One can build a single star for each data record, but also one star which contains/represents all records (→ overlapping problem)



[Inspired by J. Stasko's course]

Star Plots – Examples





(→ "small multiples" (icons))

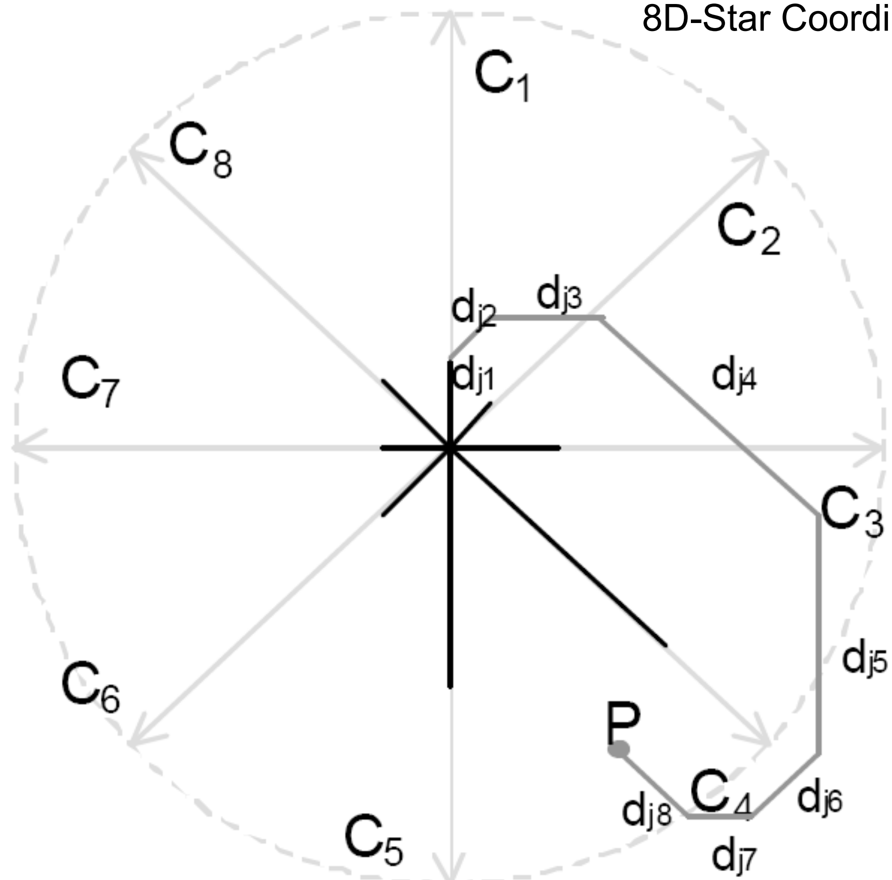
[Inspired by J. Stasko's course]

4.6.2 Axis-based Approaches

Star Coordinates

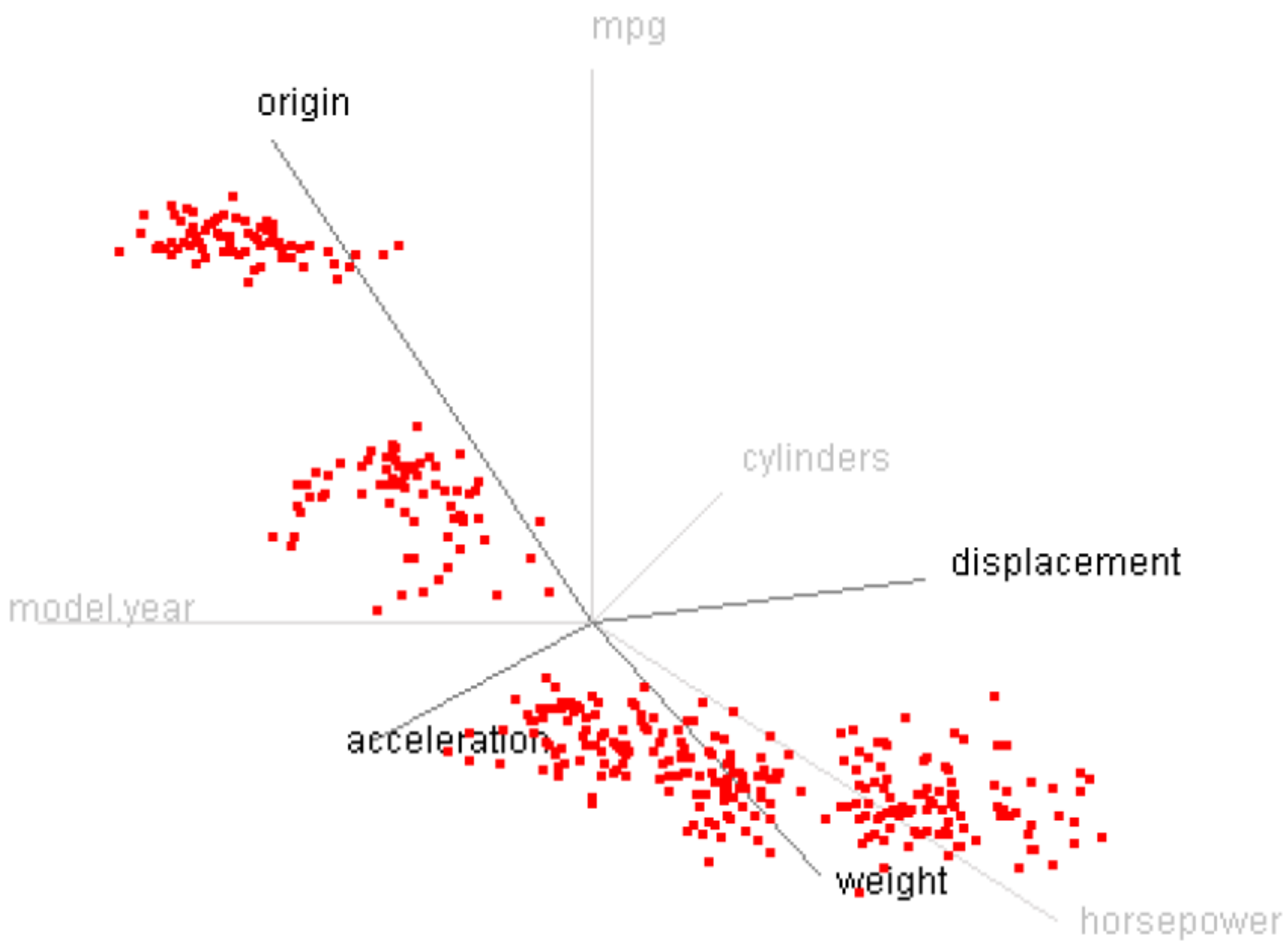
- Combination of Star Plots and Scatter Plots
- Initially, all axes have the same length
- Positioning dots like shown in the figure
- Interaction
  - Scale axes
  - Rotate axes
  - Select dots

8D-Star Coordinate



[E. Kandogan, "Star Coordinates: A Multi-dimensional Visualization Technique with Uniform Treatment of Dimensions", InfoVis 2000, Late-Breaking Hot Topics, Oct. 2000]

Star Coordinates – Examples

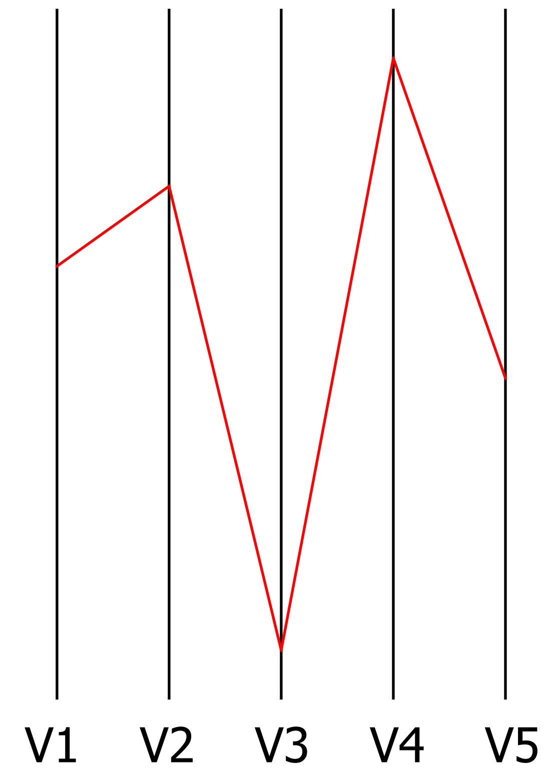


4.6.2 Axis-based Approaches

Parallel Coordinates

- If we have *n* variables, then we can draw *n* parallel axes and enter the values on the corresponding axes
- Each axis is scaled on the minimal/maximal range of all occurring values
- Each data object is represented by a polygon through all axes
- Well-known and multiple used technique

[Point-Line Duality]

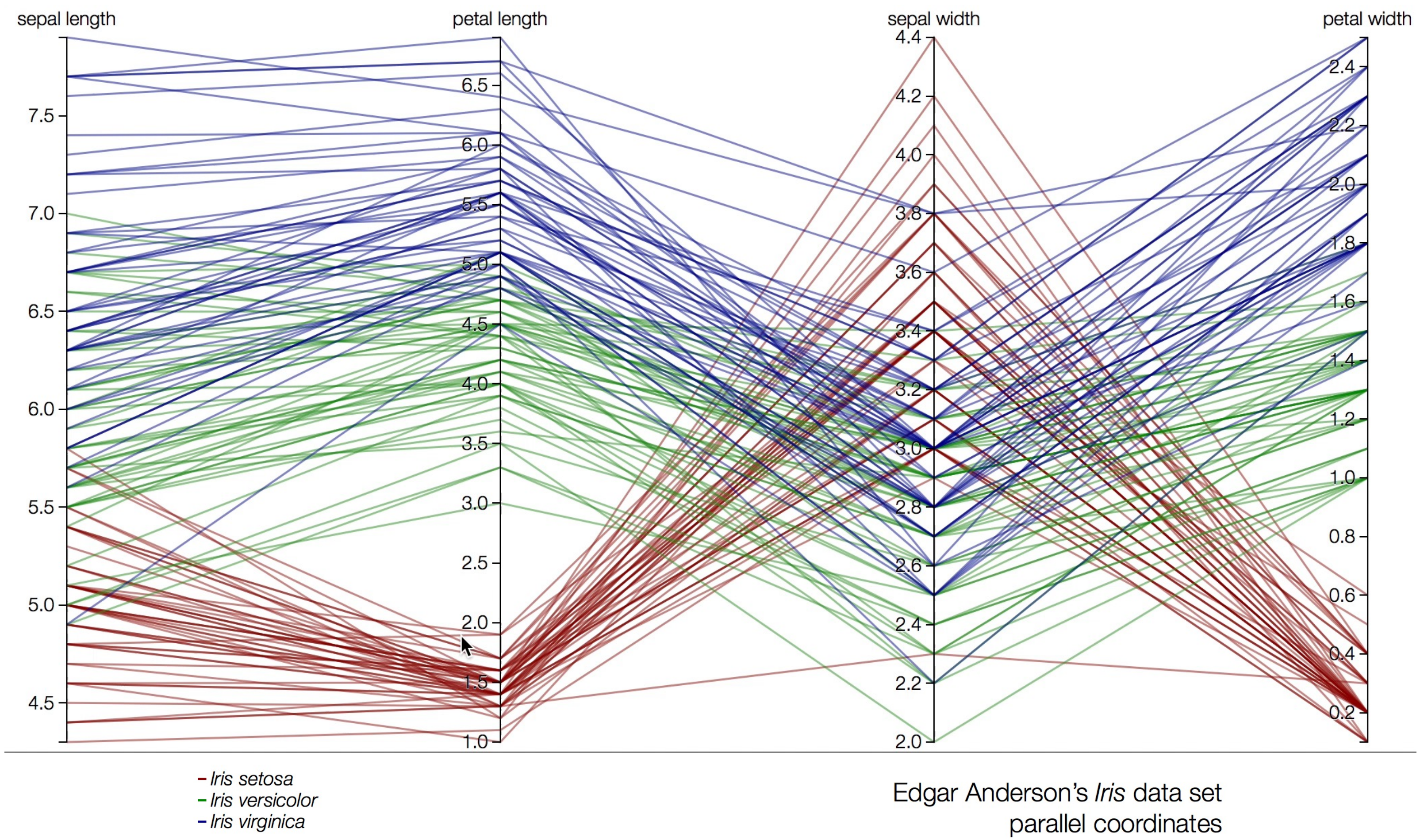


[A. Inselberg, "Multidimensional Detective", InfoVis '97, 1997]

[Inspired by J. Stasko's course]

4.6.2 Axis-based Approaches

Parallel Coordinates Plot



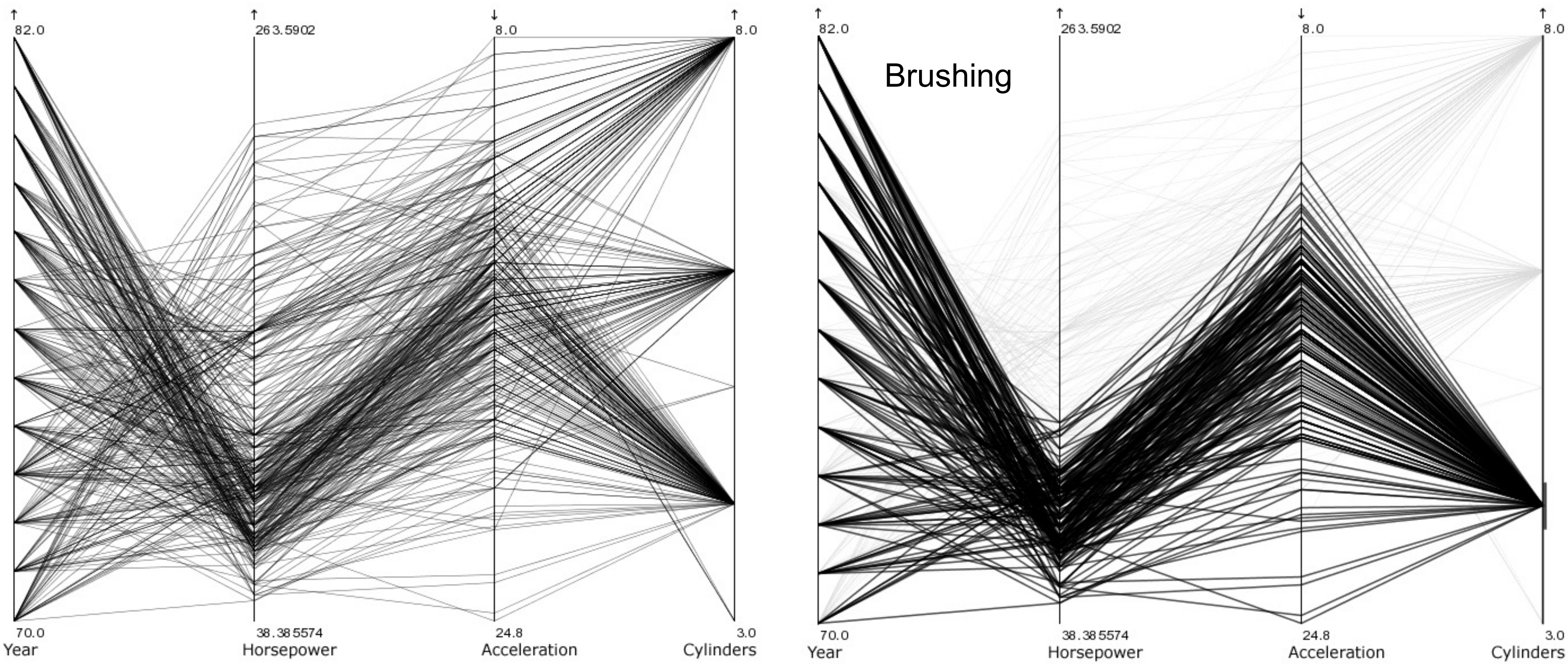
[Demo]

4.6.2 Axis-based Approaches

- Disadvantages of Parallel Coordinates
  - With many objects, there are many overlaps
  - Order of axes matters
  - The original idea is relatively inflexible
    - There are a lot of extensions!!
- To avoid the most important disadvantages, clever interaction techniques for Parallel Coordinates were developed (also F&C)
- Example application
  - Parvis

4.6.2 Axis-based Approaches

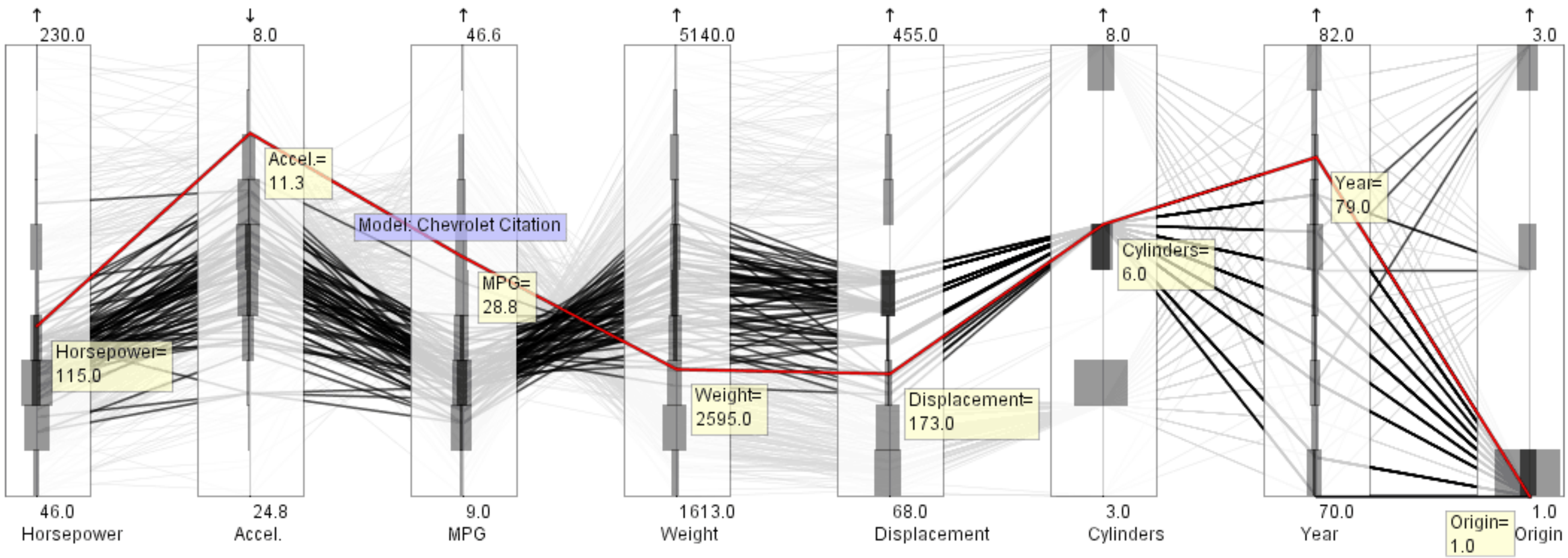
Parvis – Examples



http://www.vrvis.at

Parvis – Examples (cont.)

Brushing + Histograms

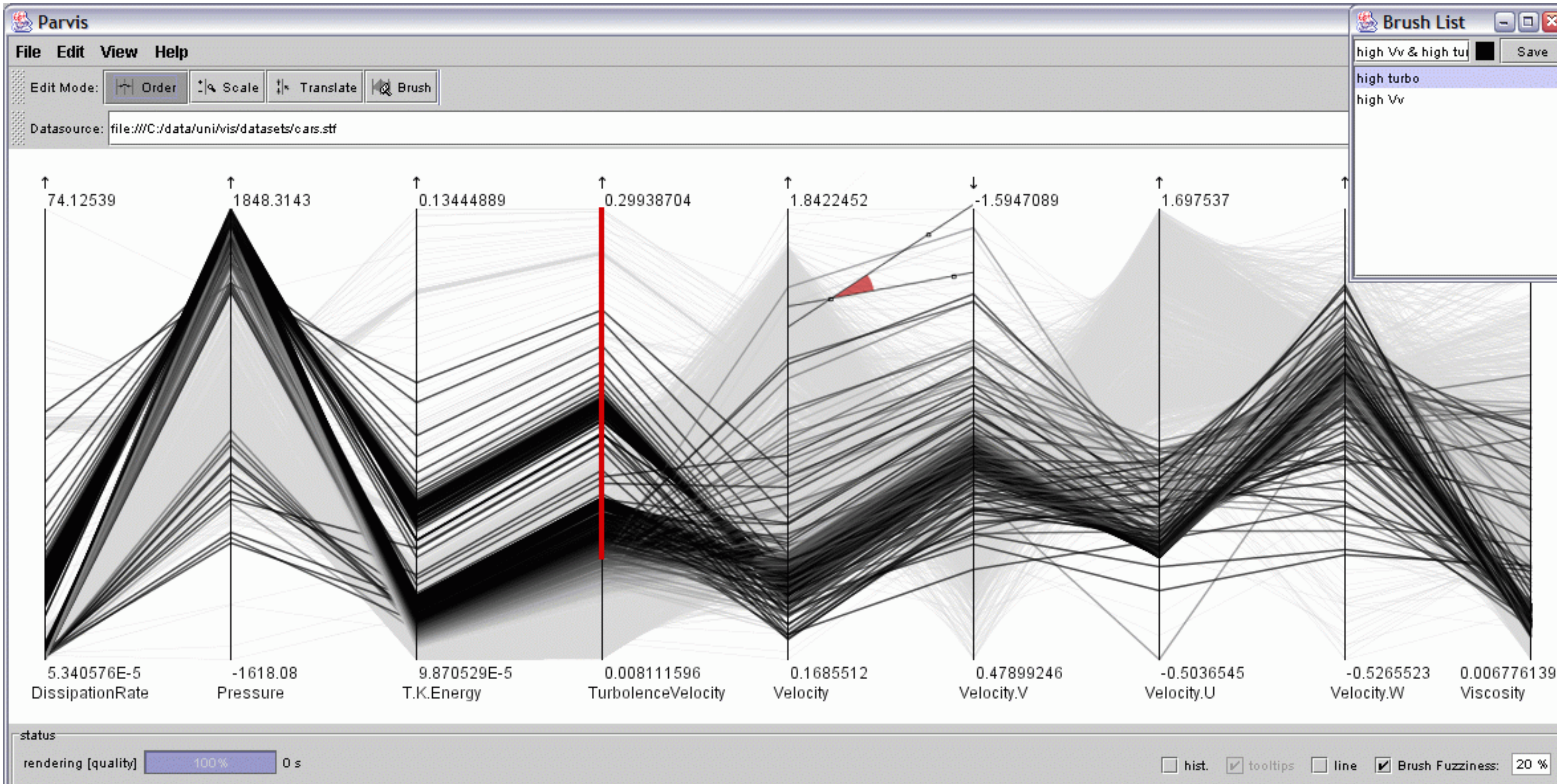


http://www.vrvis.at

4.6.2 Axis-based Approaches

Parvis – Examples (cont.)

Selection through axis ranges and angles (both red) → Focus



http://www.vrvis.at

4.6.3 Icon-based Approaches

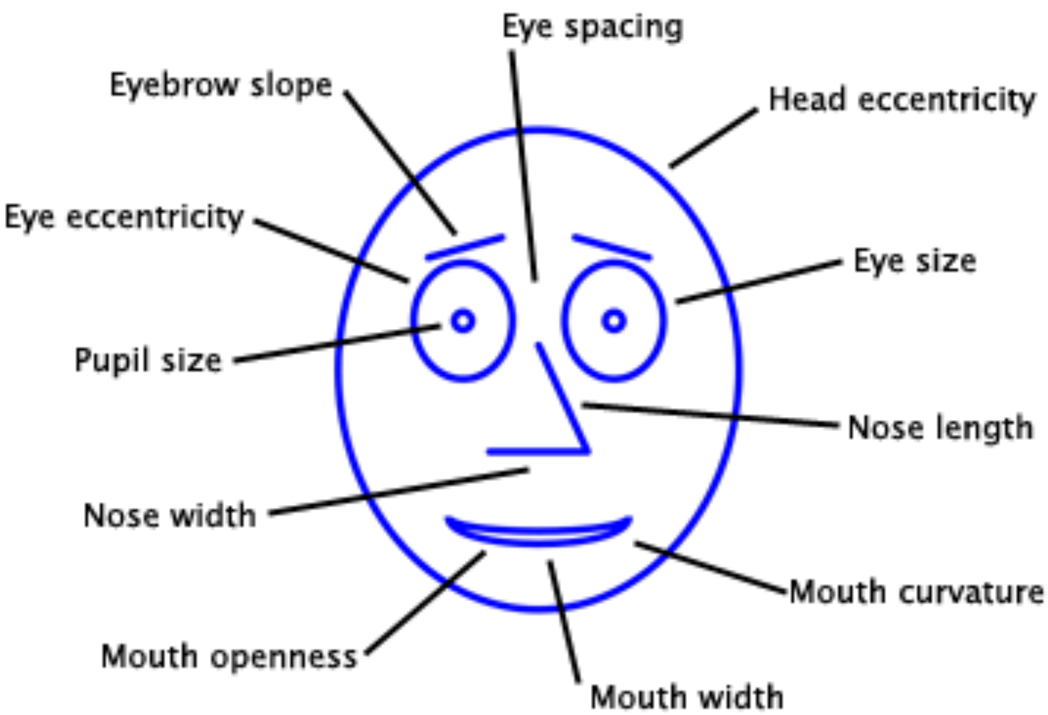
- In context of icon-based techniques, small graphical marks/elements (icons) are modified in dependency of one or several variable values
- With the help of other variable values, it is possible to specify their position on the screen
- Examples
  - Chernoff Faces
  - Stick Figures
  - Shape Coding
  - …

4.6.3 Icon-based Approaches

Chernoff Faces

- They use the human capability to differ between faces well

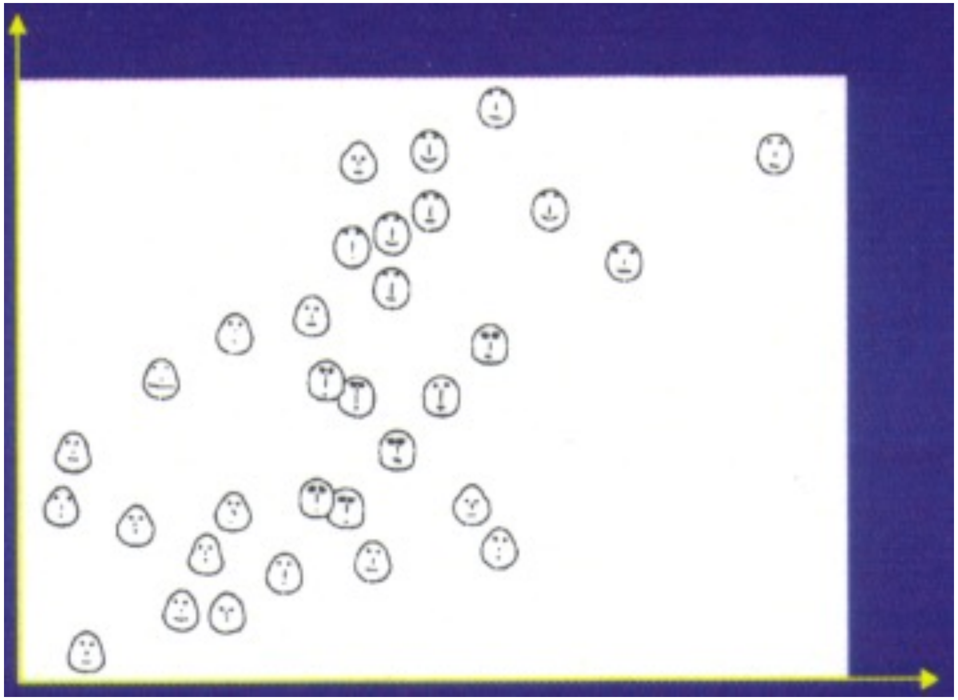
- Attributes of Chernoff Faces are form of the mouth, height and position of the eyebrows, form of the face, etc.



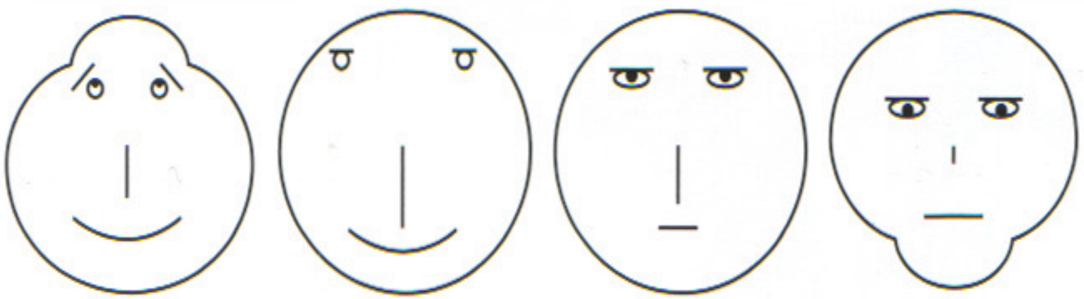
http://hesketh.com/schampeo/projects/Faces/chernoff.html (Java Demo)

4.6.3 Icon-based Approaches

Chernoff Faces – Examples

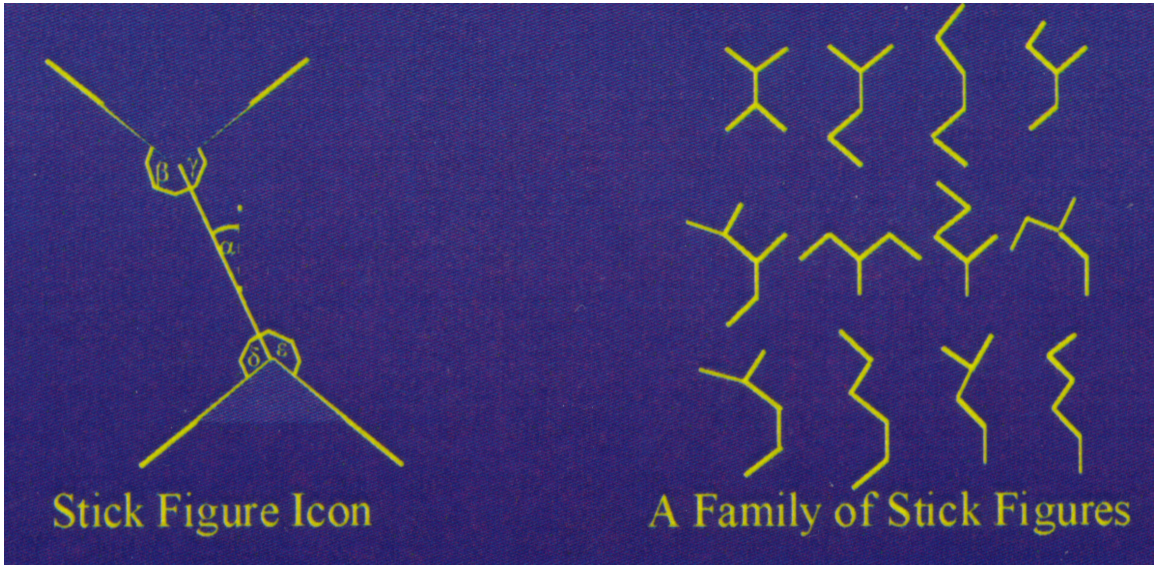


Chernoff faces, in which facial characteristics encode the values of variables



Stick Figures

- Two quantitative (or ordinal) attributes are used for positioning in the plane
- The rest of attributes are used for the angles and/or lengths of the lines

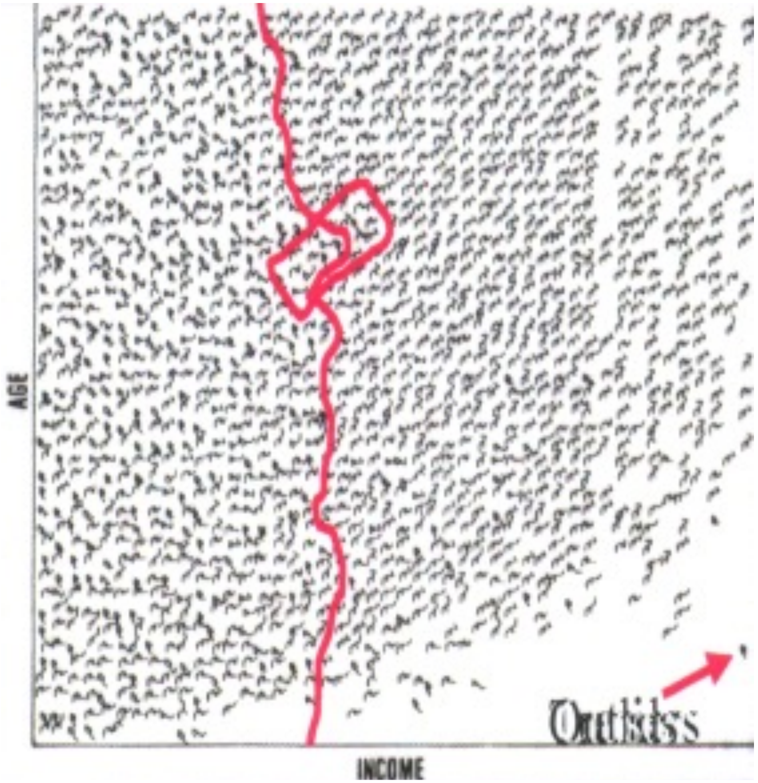


[R. M. Pickett and G. G. Grinstein, "Iconographic Displays for Visualizing Multidimensional Data," Proceedings of the 1988 IEEE Conference on Systems, Man and Cybernetics, Beijing and Shenyang, China, 1988.]

4.6.3 Icon-based Approaches

Stick Figures – Examples

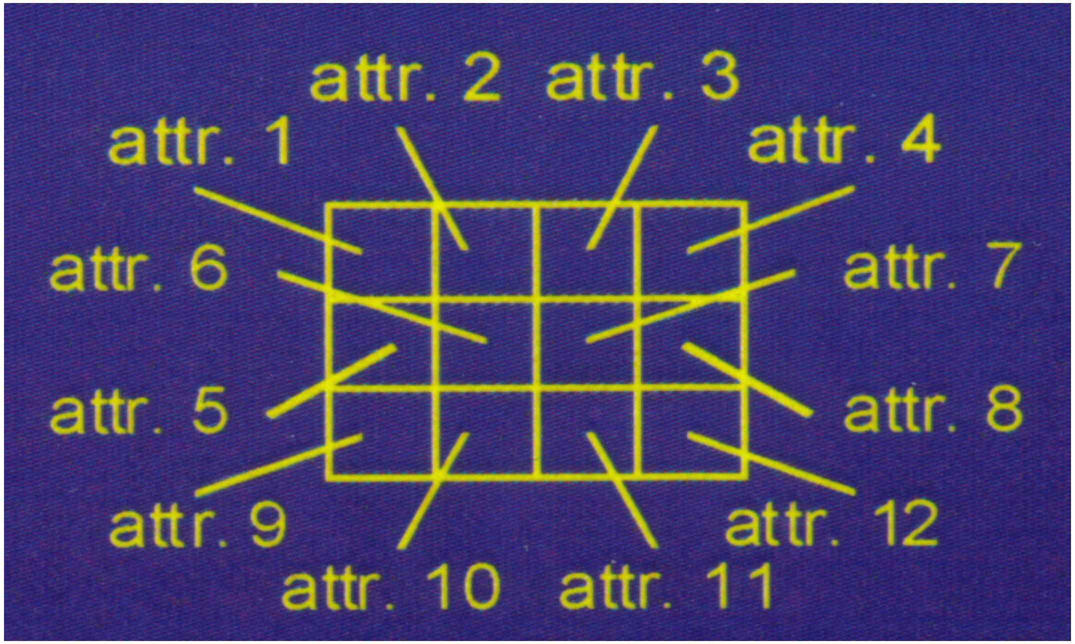
- Often used in Scientific Visualization



4.6.3 Icon-based Approaches

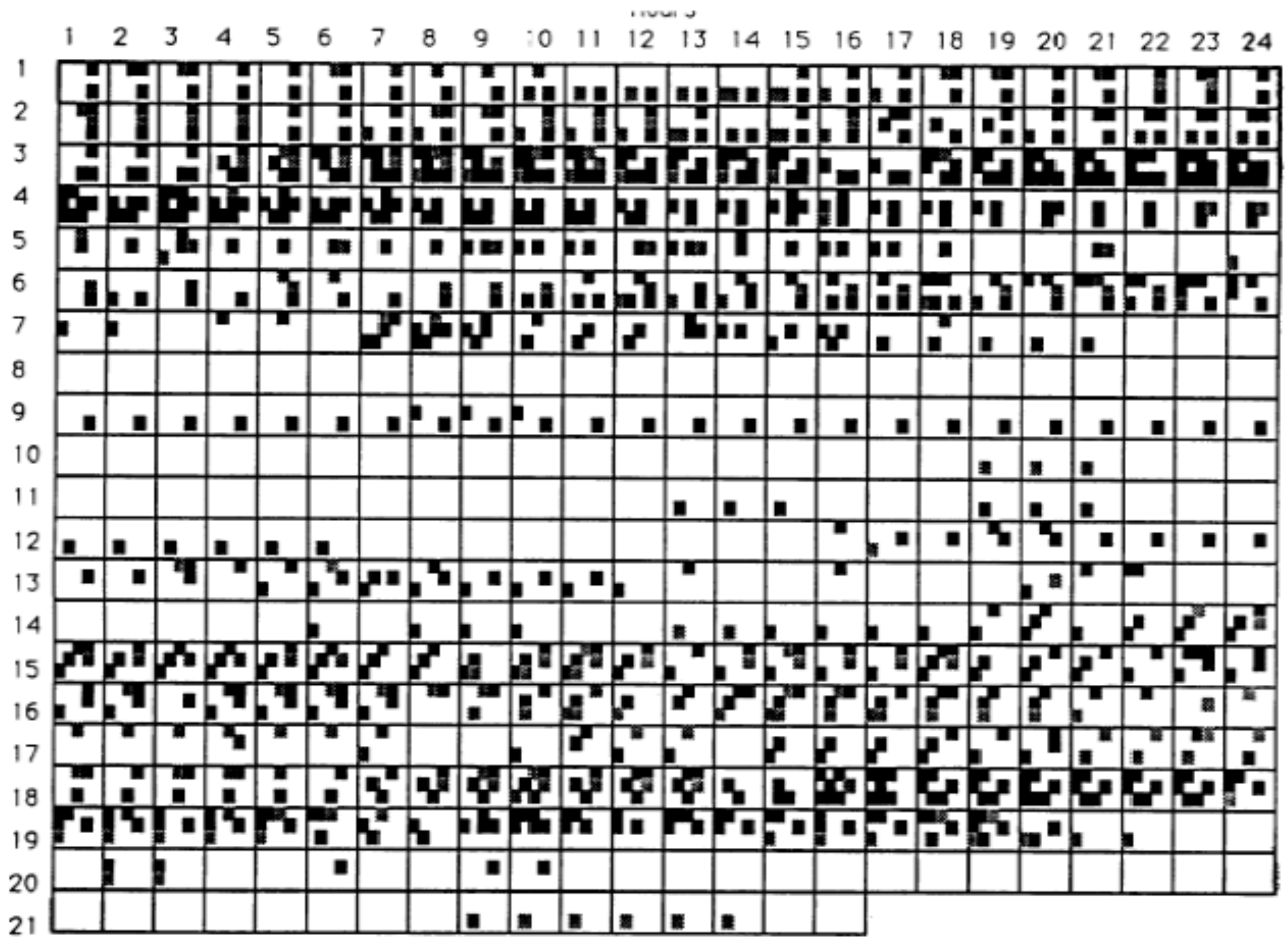
Shape Coding

- In this technique, small rectangles, that hold a specific (rectangle) part for each variable, are used to display the individual records
- These rectangles are positioned using, for example, the time (with time-series data) as ordering method



[Beddow J.: "Shape Coding of Multidimensional Data on a Microcomputer Display", Visualization 1990, San Francisco, CA, 1990, pp. 238-246]

Shape Coding – Examples



4.6.4 Pixel-based Approaches

- If we have a huge set of cases and/or attributes then we exceed the limit of the screen resolution because many pixels for each case/variable are used.
- Pixel-based techniques (*Dense Pixel Displays*) optimally dispose the display capacity by using one pixel for one tabular entry
- The color of a pixel mostly represents the value itself, and the individual values over all cases, i.e., the rows of the data table, build own areas on the screen

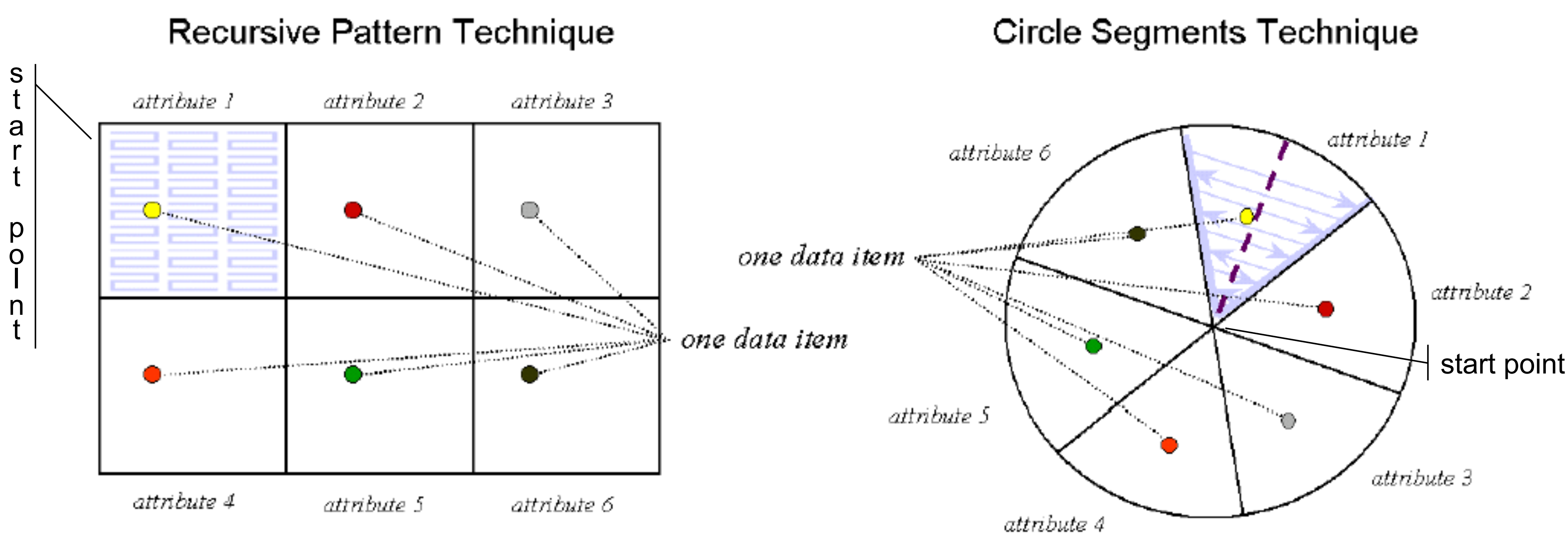
4.6.4 Pixel-based Approaches

- Questions
  - How are the pixels positioned within the areas?
  - Are there other shapes instead of rectangles?
  - How can the variables (dimensions, areas) be arranged?

- Techniques, in order to position pixels resp. areas:
  - Recursive Patterns
  - Circle segments

4.6.4 Pixel-based Approaches

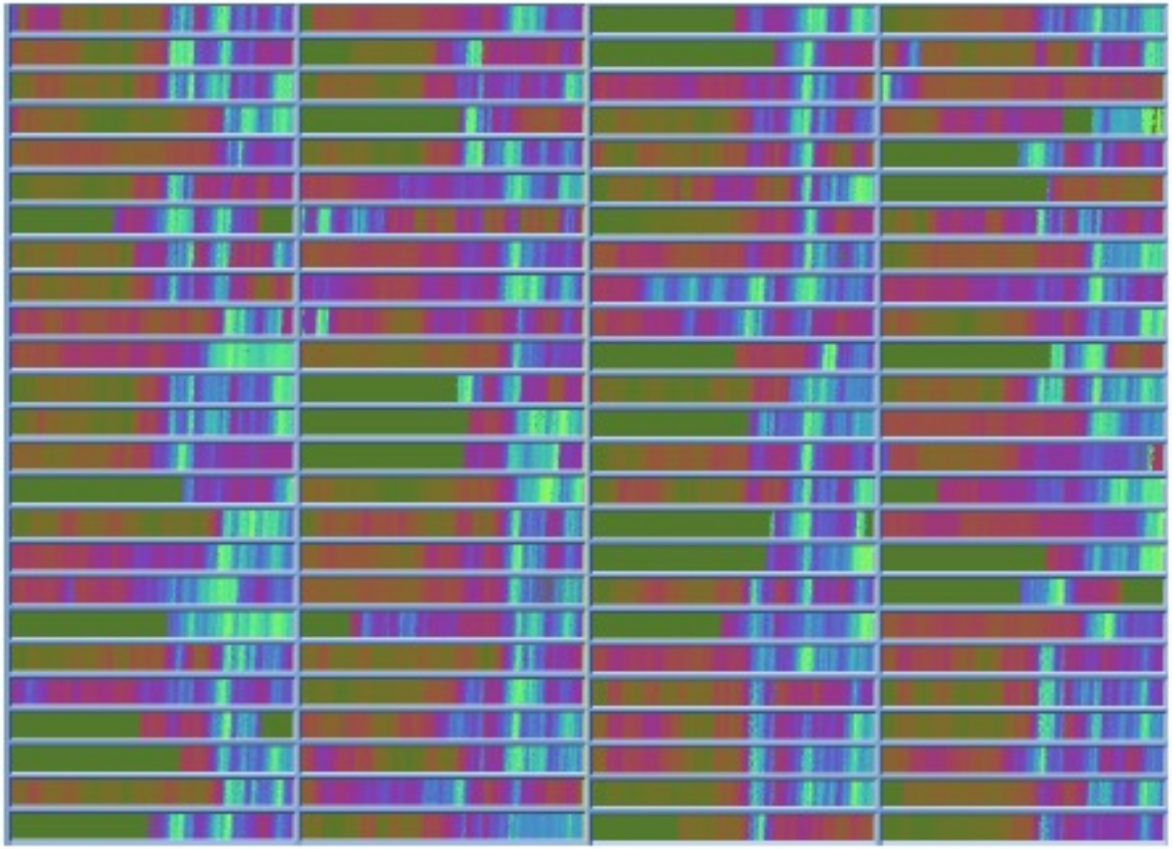
- Recursive Patterns vs. Circle Segments



[Daniel A. Keim. "Information Visualization and Visual Data Mining," IEEE Transactions on Visualization and Computer Graphics, vol. 08, no. 1, pp. 1-8, January-March, 2002.]

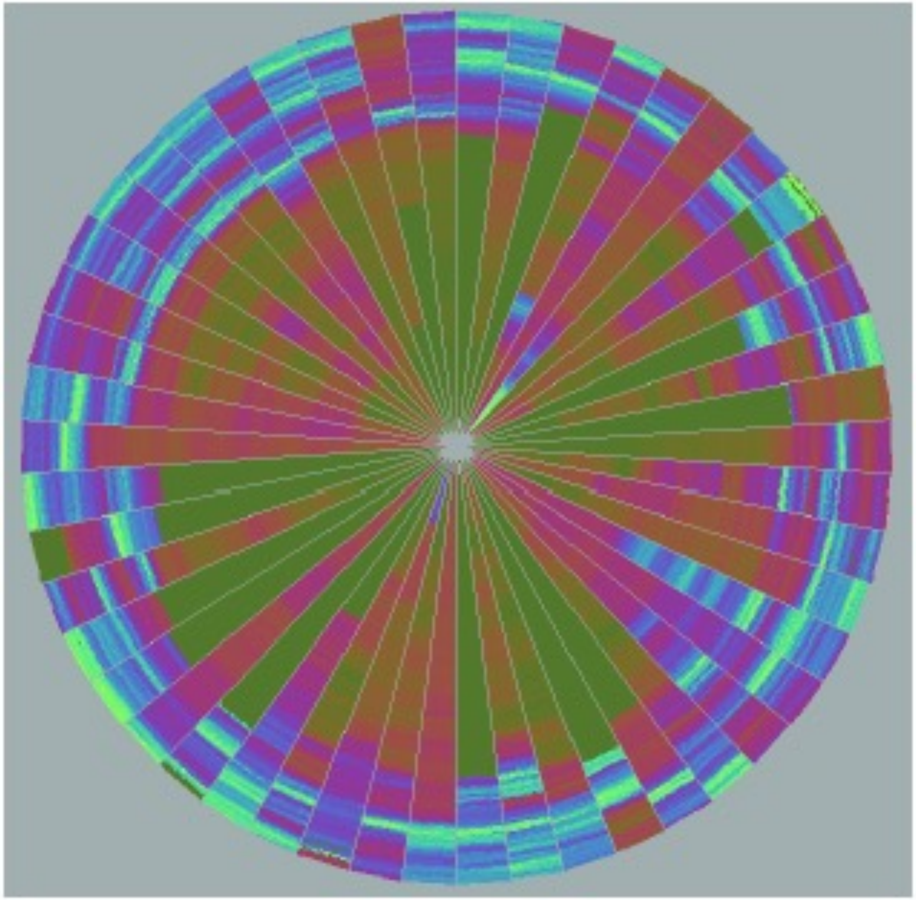
Recursive Patterns – Example

- Visualized are the prices of 100 stock values in the stock index of the Frankfurter Allgemeinen Zeitung (FAZ) over a time of 20 years
- → Only *one* attribute in this visualization that changes over time



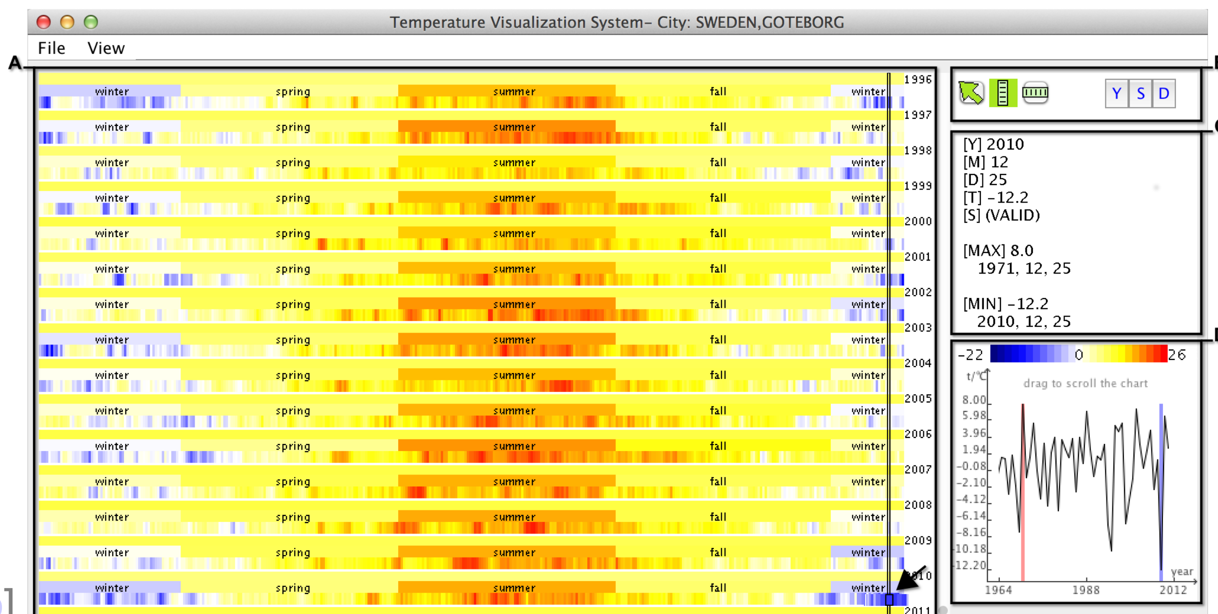
Circle Segments – Example

- Visualized are the prices of 50 stock values in the FAZ stock index over a period of 20 years
- The positioning process starts in the center of the circle and proceeds in direction of the periphery
- „Older“ values are closely related in the center



Example: Multi-Scale Trend Visualization

- Temperature data over the past >100 years
- Multiscale: days/seasons/ years, cities, ….
- Several techniques implemented: pixel-based representation, plot, magic lens, missing data visualization, …



[Video]

https://ep.liu.se/konferensartikel.aspx?series=ecp&issue=106&Article_No=13

4.7 Integrating Computations

- The approaches discussed above focus on *visual representations* rather than computational steps
  - They are usually *linear* with regard to the mapping and *deterministic* with regard to the resulting layout
- Some other strategies for n-variate data which may go beyond the traditional scope of InfoVis → often rely on optimization / iterative computations [→ TNM098 (partly)]
  - Force-based approaches
    - Models inspired by Graph Drawing, RadViz, …
  - Dimensionality reduction
    - PCA, MDS, t-SNE, …

4.7.1 Force-based Approaches

- Inspired by physical simulations and very popular in Graph Drawing
- Basic idea:
  - Represent the visual items as dots or glyphs in a 2D/3D space
  - Define attractive and repulsive forces based on the data attribute values and/or connectivity (e.g., two items should not overlap → push them away from each other!)
  - On each iteration, the 2D/3D positions of visual items are updated according to the model
  - Run the simulation iteratively until the convergence or for a fixed number of steps

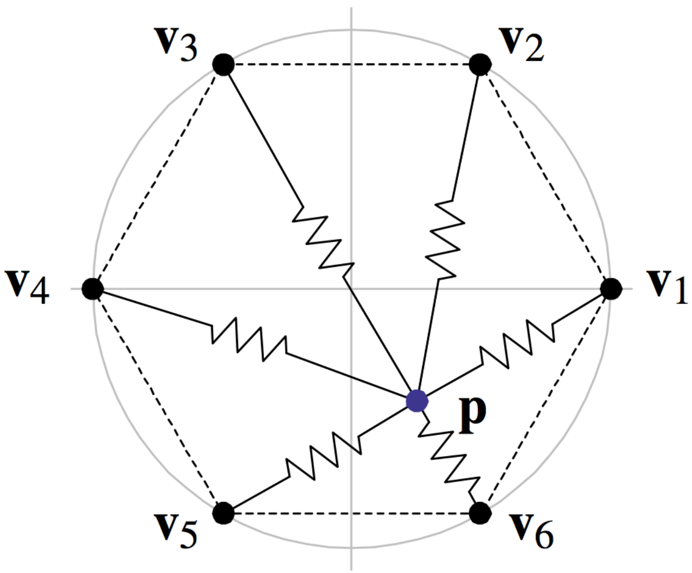
4.7.1 Force-based Approaches

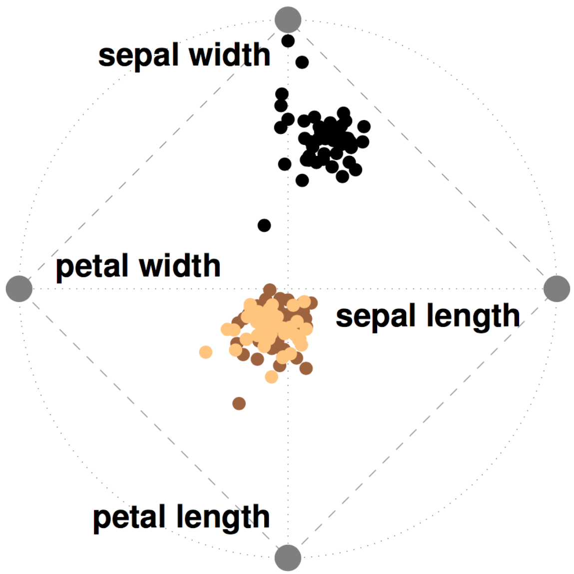
- Pros:
  - Rather easy to implement
  - Rather easy to explain & understand (also, can be animated easily!)
  - May reveal clusters/groups (with a proper model)
  - Easy to avoid issues such as overplotting

- Cons:
  - Nondeterministic: the results *may* vary a lot
  - Performance issues
  - Configuration / parameter adjustment issues

4.7.1 Force-based Approaches

- RadViz
  - Force-based model which uses a radial layout
  - Items are attracted towards *anchor points* representing dimensions





[M. Rubio-Sánchez et al., "A Comparative Study between RadViz and Star Coordinates," IEEE Transactions on Visualization and Computer Graphics, 22(1):619-628, 2016]

4.7.2 Dimensionality Reduction

- Based on computational methods and very popular in Machine Learning, Data Mining, etc.
- Basic idea:
  - Use either data attribute values or *distances* between data items (depends on the individual DR method)
  - Compute a projection from *n*D to 2D/3D which is optimal with regard to certain criteria such as distances between items
    - The actual computation process may vary from linear (e.g., PCA) to non-linear (e.g., t-SNE) methods
    - Note: new dimensions are generated instead of selecting the existing ones!
  - Represent the visual items as dots or glyphs in the resulting 2D/3D space

4.7.2 Dimensionality Reduction

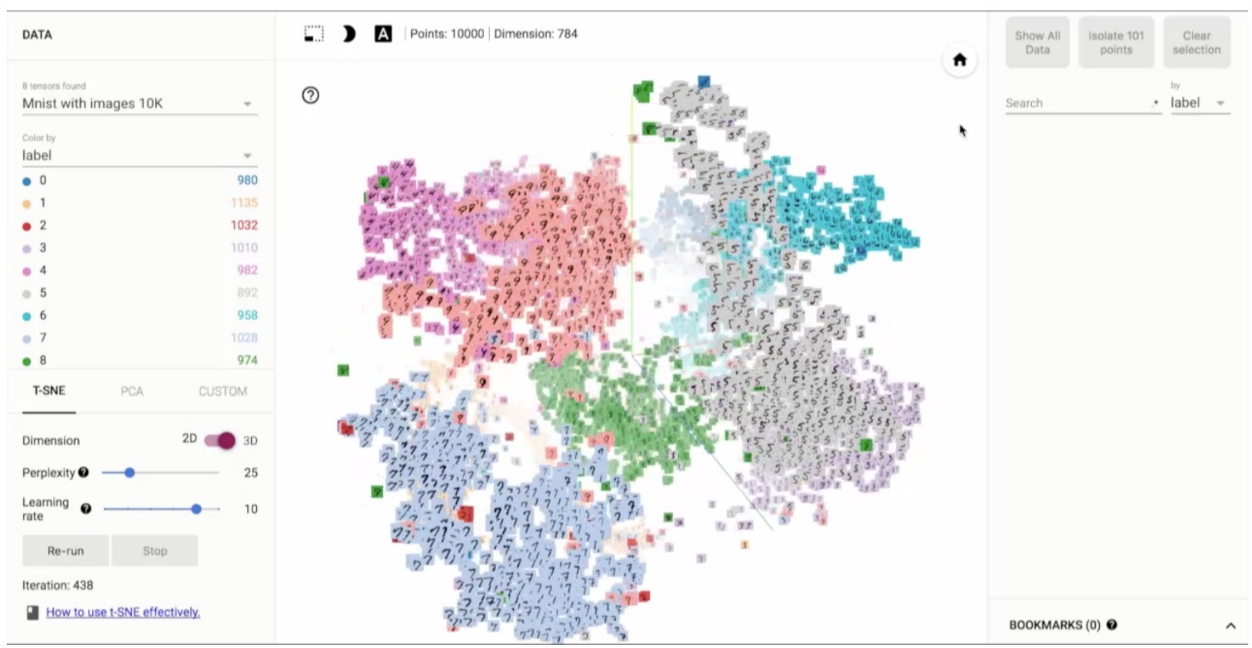
- Pros:
  - The *representation* and *the basic idea* are rather easy to explain
  - May reveal clusters/groups (with a proper model)
  - Some efficient implementations are available
- Cons:
  - The *underlying method* may be very difficult to understand & implement
  - May be nondeterministic (depending on the DR method)
  - Configuration / parameter adjustment issues

4.7.2 Dimensionality Reduction

- Several examples and links to tutorials/demos:
  - Principal Component Analysis (**PCA**)
    - http://setosa.io/ev/principal-component-analysis/
    - https://sebastianraschka.com/Articles/2015_pca_in_3_steps.html
  - Multidimensional Scaling (**MDS**)
    - http://en.wikipedia.org/wiki/Multidimensional_scaling
    - http://scikit-learn.org/stable/auto_examples/manifold/plot_mds.html
  - t-Distributed Stochastic Neighbor Embedding (**t-SNE**)
    - http://lvdmaaten.github.io/tsne/
    - https://distill.pub/2016/misread-tsne/

4.7.2 Dimensionality Reduction

- Applications of DR methods:
  - Visualization of the MNIST data set
    - http://colah.github.io/posts/2014-10-Visualizing-MNIST/
  - Visualization of embeddings in TensorBoard/TensorFlow
    - http://tensorflow.org/get_started/embedding_viz



http://tensorflow.org/get_started/embedding_viz