CSE 564
Visualization & Visual Analytics

High-Dimensional Data

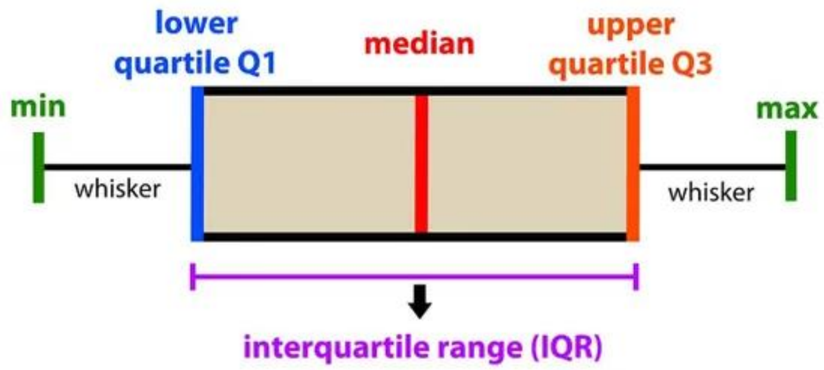
Klaus Mueller

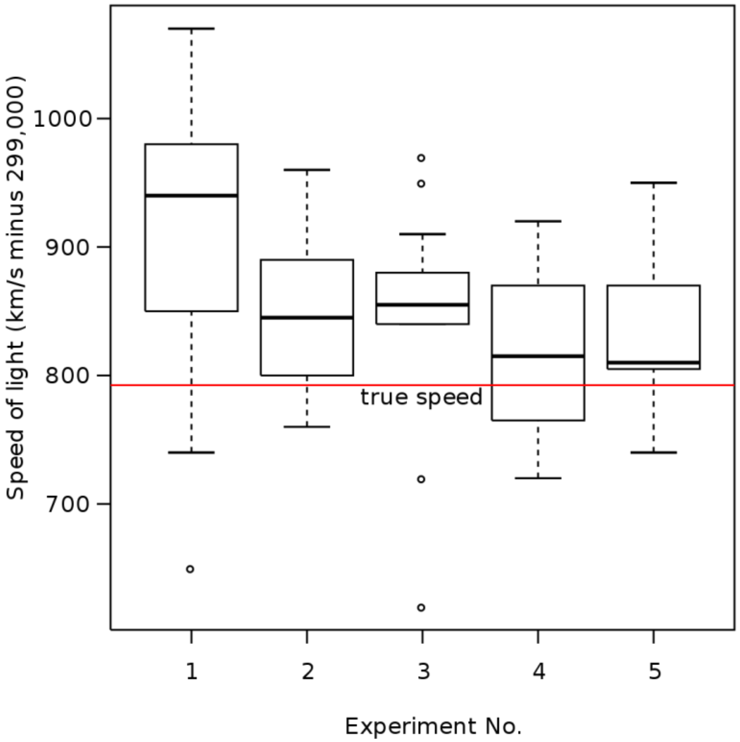
Computer Science Department
Stony Brook University

Lecture	Topic	Projects
1	Intro, schedule, and logistics	
2	Applications of visual analytics	
3	Basic tasks, data types	Project #1 out
4	Data assimilation and preparation	
5	Introduction to D3	
6	Bias in visualization	
7	Data reduction and dimension reduction	
8	Data reduction and dimension reduction	Project #2(a) out
9	Visual perception and cognition	
10	Visual design and aesthetics	
11	Cluster analysis: numerical data	
12	Cluster analysis: categorical data	Project #2(b) out
13	High-dimensional data visualization	
14	Dimensionality reduction and embedding methods	
15	Principles of interaction	
16	Midterm #1	
17	Visual analytics	Final project proposal call out
18	The visual sense making process	
19	Maps	
20	Visualization of hierarchies	Final project proposal due
21	Visualization of time-varying and time-series data	
22	Foundations of scientific and medical visualization	
23	Volume rendering	Project 3 out
24	Scientific and medical visualization	Final Project preliminary report due
25	Visual analytics system design and evaluation	
26	Memorable visualization and embellishments	
27	Infographics design	
28	Midterm #2	

Interlude – Box Plots

You may have heard about box plots





Tend to be bewildering to many

- hard to interpret

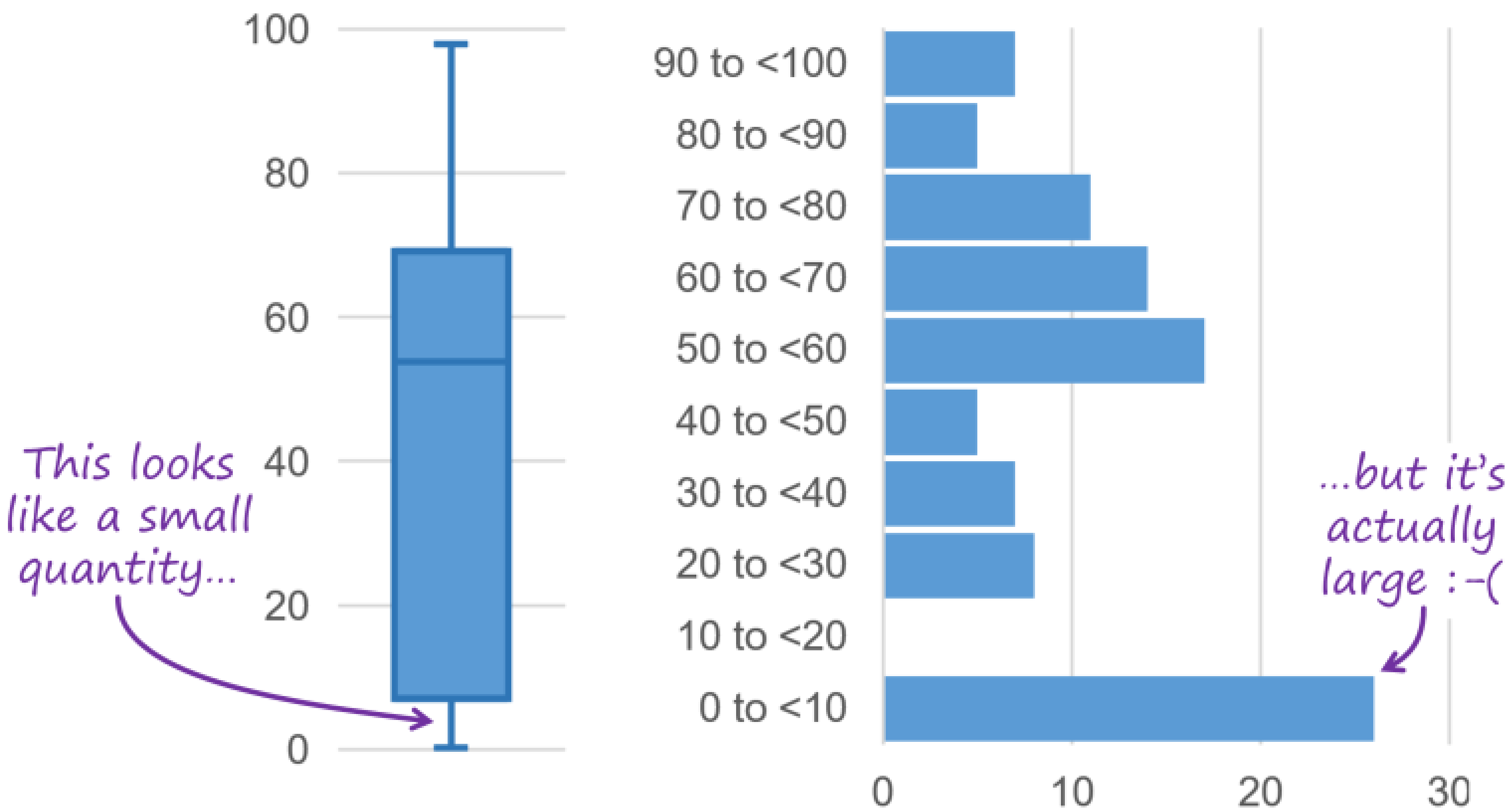
They can also give the wrong representation of data

- assume normal distributed data

Box Plots

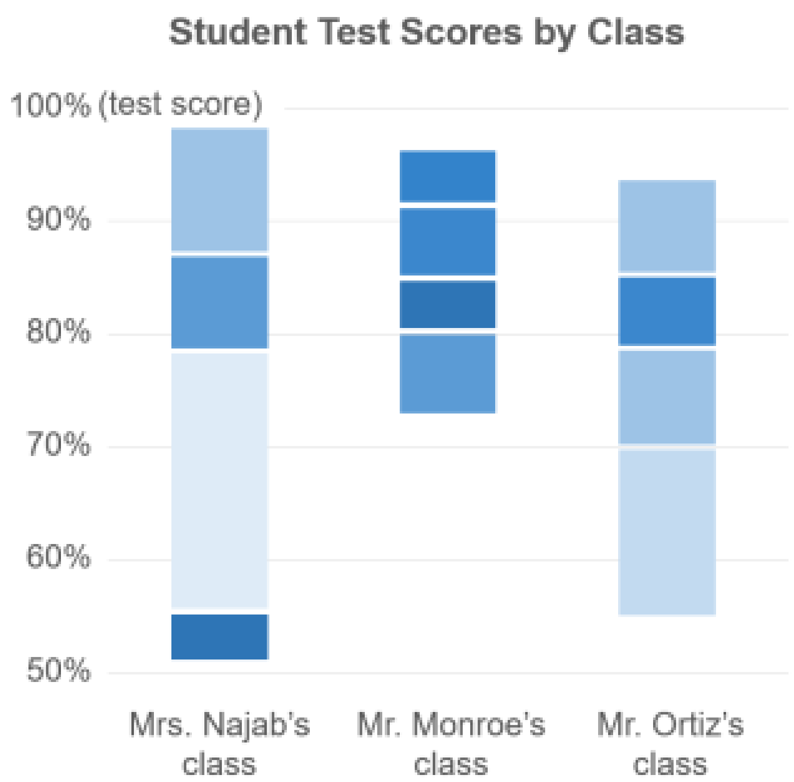
Non-normal distributed data give “wrong” box plots

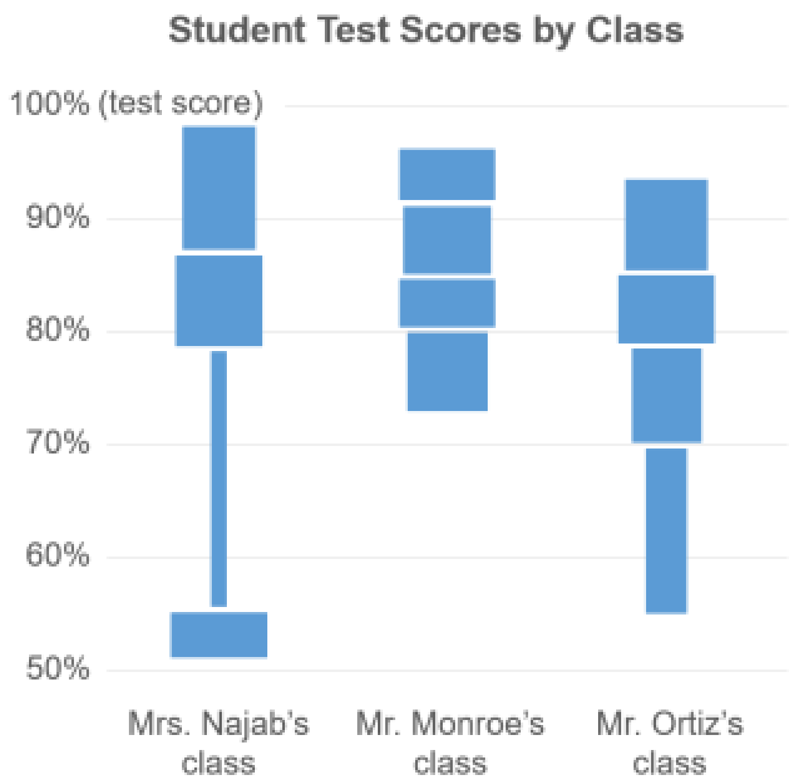
- shown here: data on student test scores



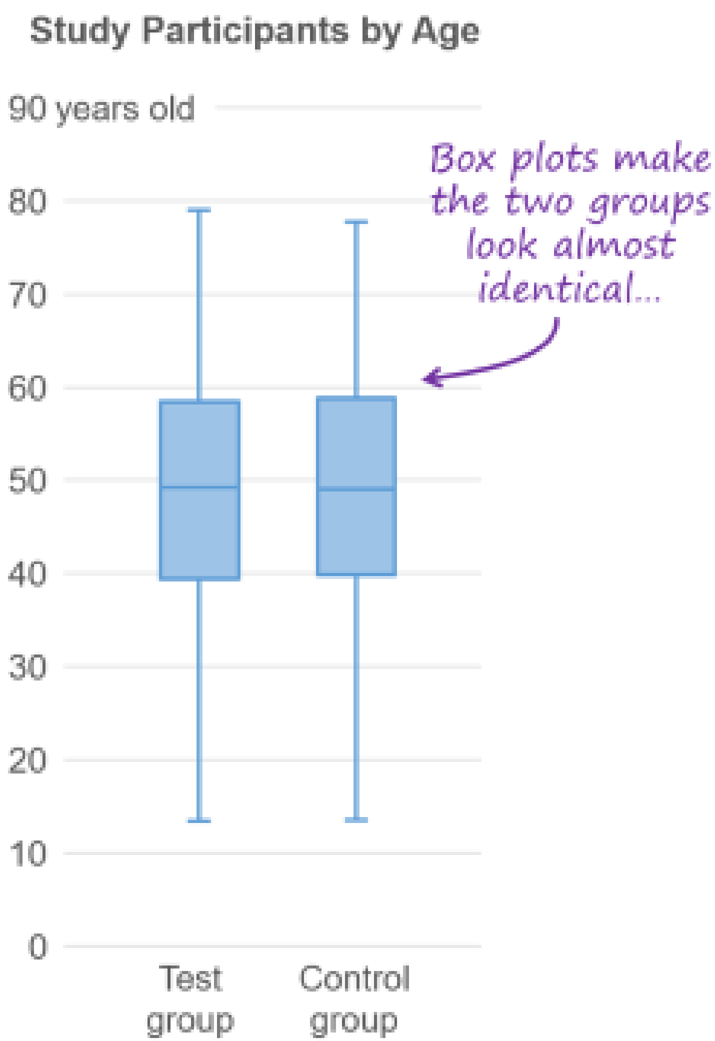
Density Plots

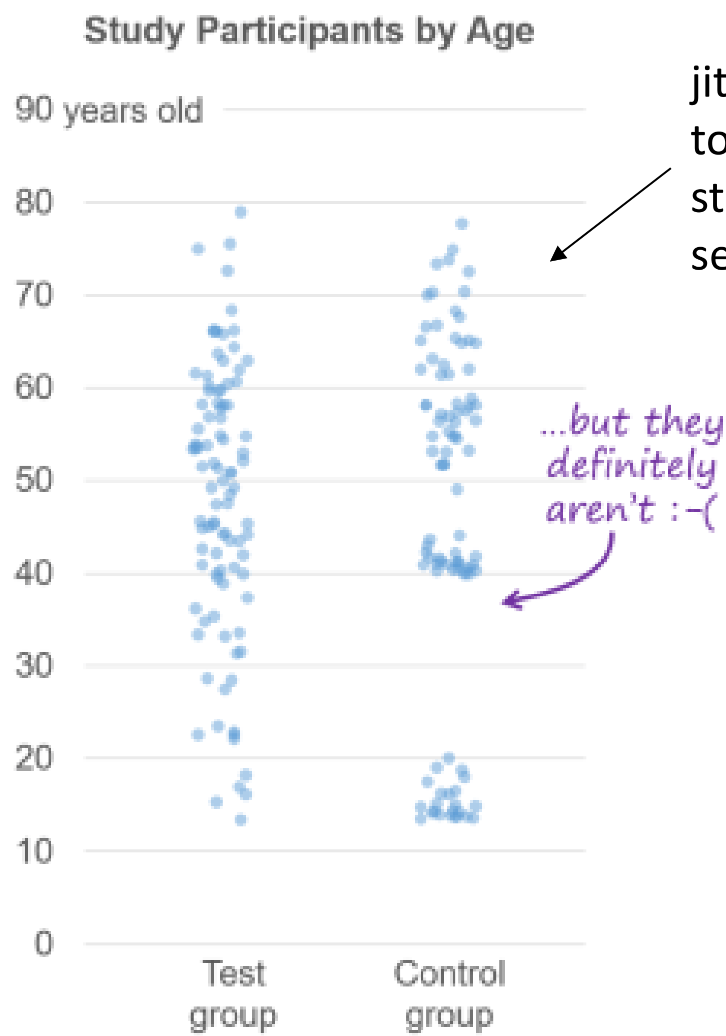
Same data than last side, multiple classes





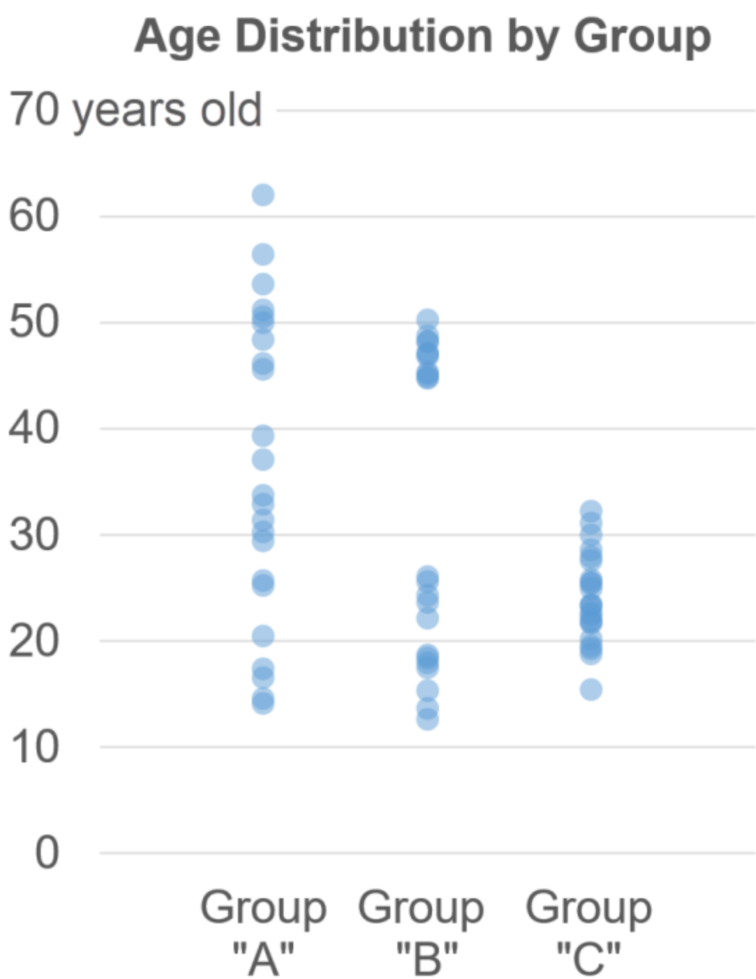
Strip Plots

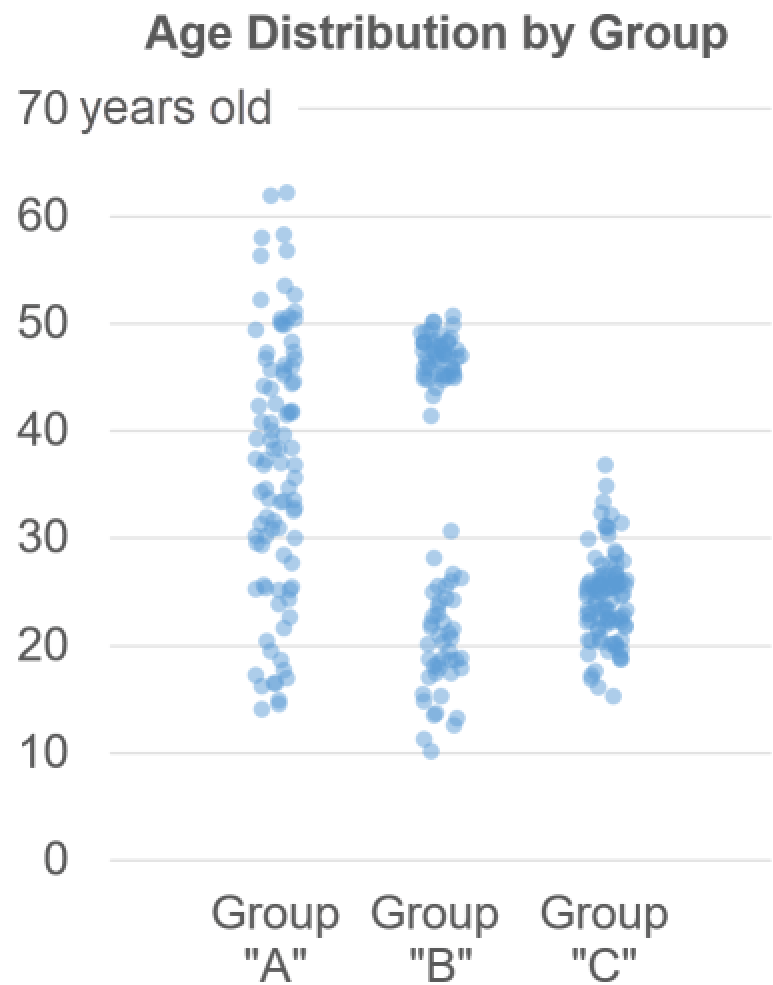




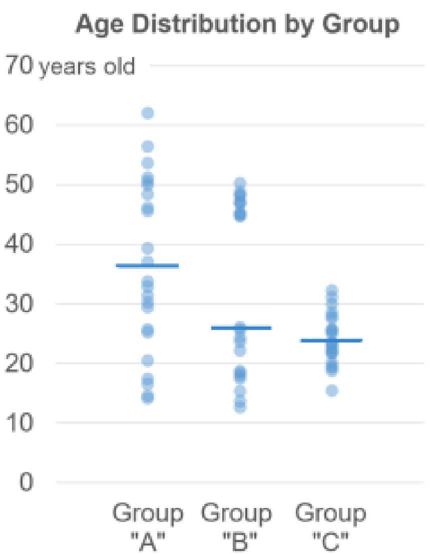
jittered points to make the stats easier to see

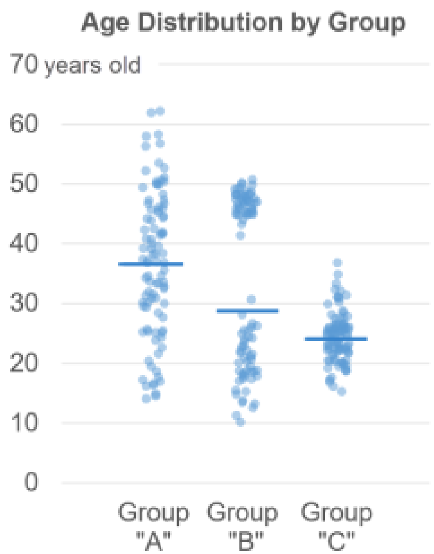
Semitransparent vs. Jittering

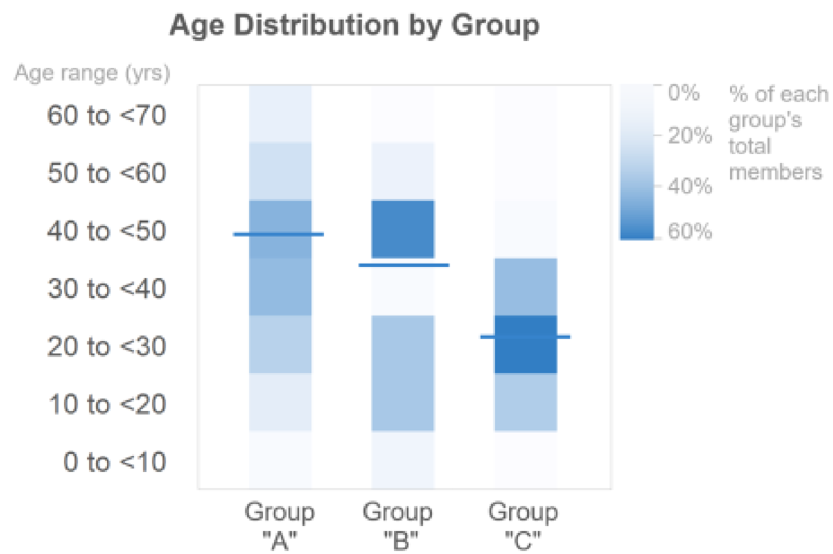




Comparison



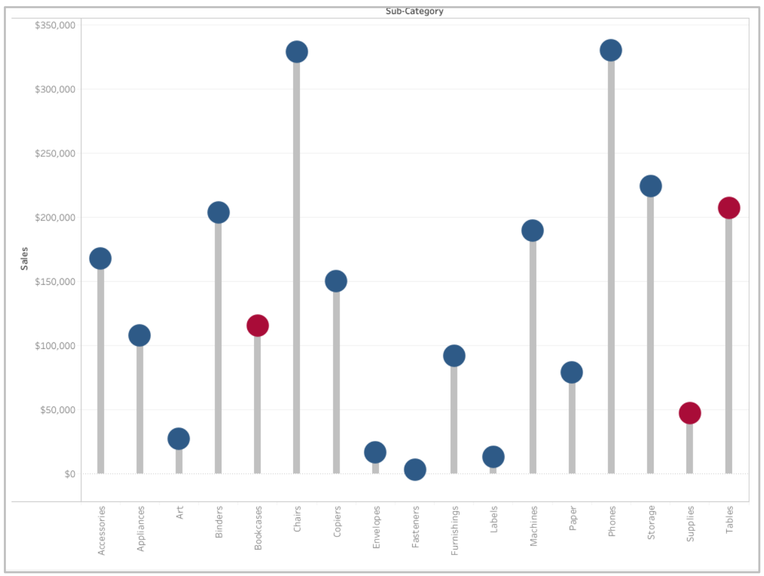


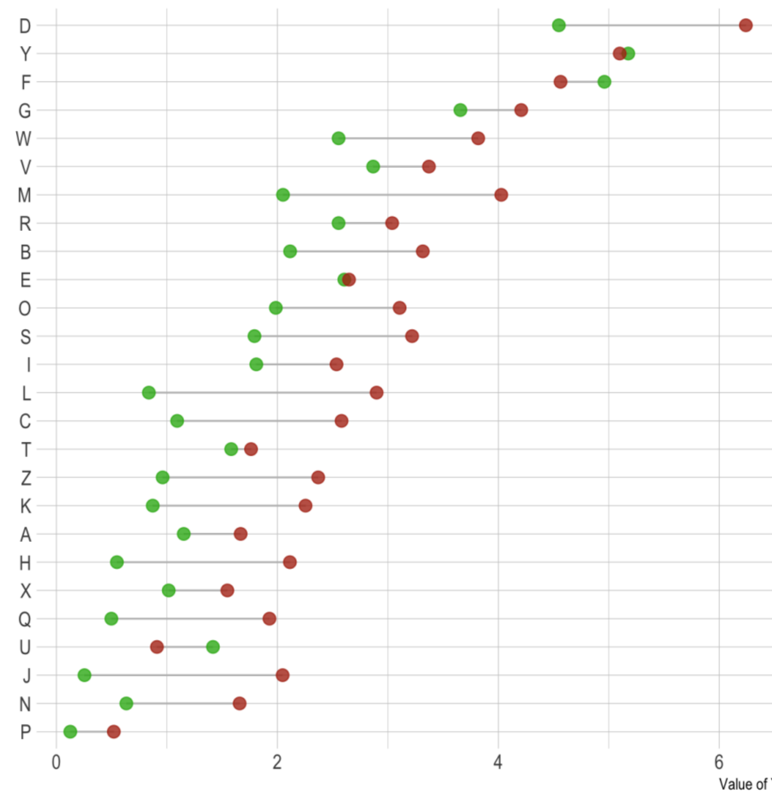


With median lines

Read more here:
https://nightingaledvs.com/ive-stopped-using-box-plots-should-you/

Lollipop Charts





makes it easier to see and compare positions than scatter plots

Rectangular Dataset

One data item

The variables or *features*
→ the attributes or properties we measured

The data items or *feature vectors*
→ the samples (observations) we obtained from the population of all instances

	A	B	C	D	E	F	G	H	I
1	**Name**	**Country**	**Miles Per Gallon**	**Accceleration**	**Horsepower**	**weight**	**cylinders**	**year**	**price**
2	Volkswagen Rabbit Dl	Germany	43,1	21,5	48	1985	4	78	2400
3	Ford Fiesta	Germany	36,1	14,4	66	1800	4	78	1900
4	Mazda GLC Deluxe	Japan	32,8	19,4	52	1985	4	78	2200
5	Datsun B210 GX	Japan	39,4	18,6	70	2070	4	78	2725
6	Honda Civic CVCC	Japan	36,1	16,4	60	1800	4	78	2250
7	Oldsmobile Cutlass	USA	19,9	15,5	110	3365	8	78	3300
8	Dodge Diplomat	USA	19,4	13,2	140	3735	8	78	3125
9	Mercury Monarch	USA	20,2	12,8	139	3570	8	78	2850
10	Pontiac Phoenix	USA	19,2	19,2	105	3535	6	78	2800
11	Chevrolet Malibu	USA	20,5	18,2	95	3155	6	78	3275
12	Ford Fairmont A	USA	20,2	15,8	85	2965	6	78	2375
13	Ford Fairmont M	USA	25,1	15,4	88	2720	4	78	2275
14	Plymouth Volare	USA	20,5	17,2	100	3430	6	78	2700
15	AMC Concord	USA	19,4	17,2	90	3210	6	78	2300
16	Buick Century	USA	20,6	15,8	105	3380	6	78	3300
17	Mercury Zephyr	USA	20,8	16,7	85	3070	6	78	2425
18	Dodge Aspen	USA	18,6	18,7	110	3620	6	78	2700
19	AMC Concord D1	USA	18,1	15,1	120	3410	6	78	2425
20	Chevrolet MonteCarlo	USA	19,2	13,2	145	3425	8	78	3900
21	Buick RegalTurbo	USA	17,7	13,4	165	3445	6	78	4400
22	Ford Futura	Germany	18,1	11,2	139	3205	8	78	2525
23	Dodge Magnum XE	USA	17,5	13,7	140	4080	8	78	3000
24	Chevrolet Chevette	USA	30	16,5	68	2155	4	78	2100

Understanding High-D Objects

Feature vectors are typically high dimensional

- this means, they have many elements
- high dimensional space is tricky
- most people do not understand it
- why is that?

- well, because you don't learn to see high-D when your vision system develops

Object permanence (Jean Piaget)

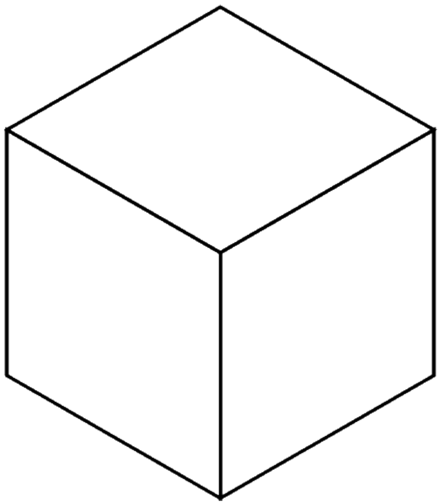
- the ability to create mental pictures or remember objects and people you have previously seen
- thought to be a vital precursor to creativity and abstract thinking

High-D Space is Tricky

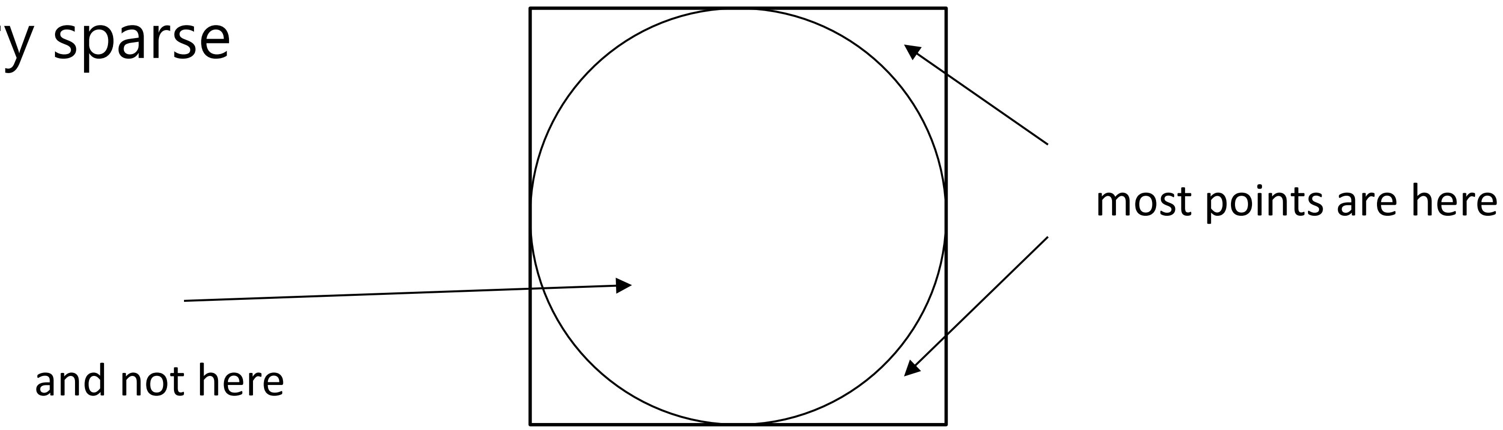
The curse of dimensionality

As $n \to \infty$

- Cube: side length l, diagonal d, volume V
- $V \to \infty$ for $l > 1$
- $V \to 0$ for $l < 1$
- $V = 1$ for $l = 1$
- $d \to \infty$

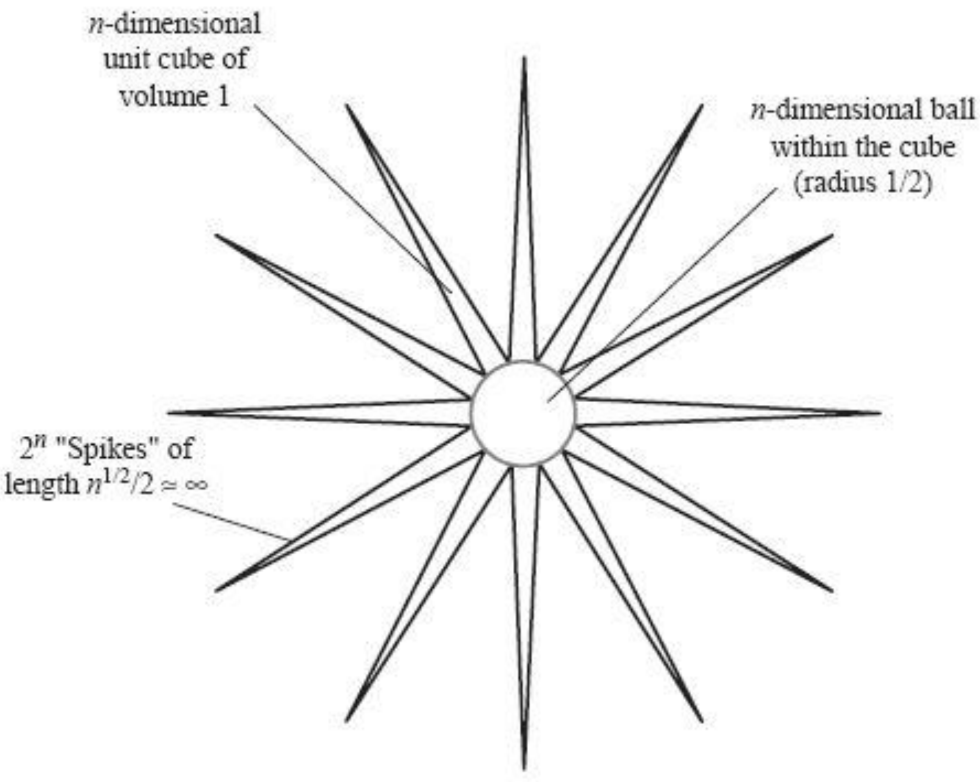


and very sparse



High-D Space is Tricky

Essentially hypercube is like a “hedgehog”



Curse of Dimensionality

Points are all at about the same distance from one another

- concentration of distances
- fundamental equation (Bellman, ‘61)

$$\lim_{n\to\infty} \frac{Dist_{\max} - Dist_{\min}}{Dist_{\min}} \to 0$$

- so as *n* increases, it is impossible to distinguish two points by (Euclidian) distance
  - unless these points are in the same cluster of points

Sparseness Demonstration

Space gets extremely sparse

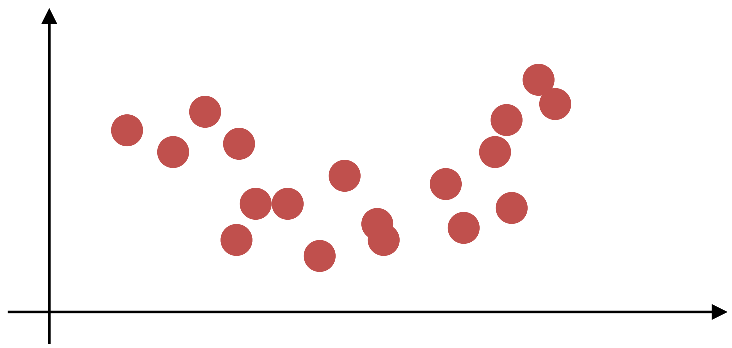
- with every extra dimension points get pulled apart further
- distances become meaningless

Sparseness Demonstration

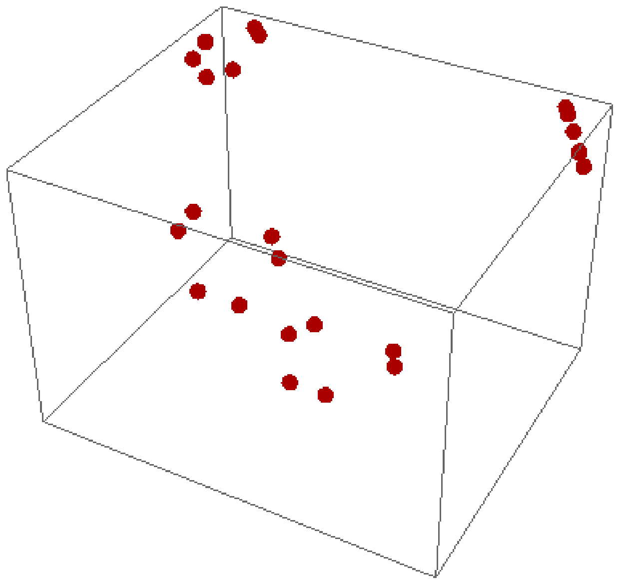
Space gets extremely sparse

- with every extra dimension points get pulled apart further
- distances become meaningless

1D – points are very close



2D – points spread apart



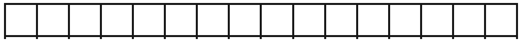
3D – getting even sparser

4D, 5D, … – sparseness grows further

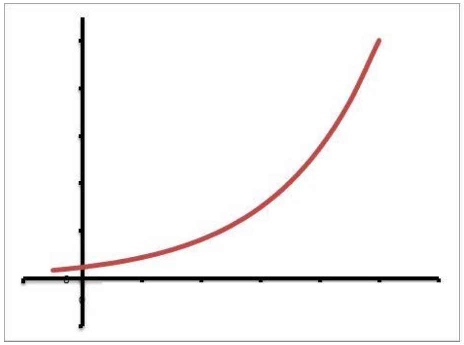
Space and Memory Management

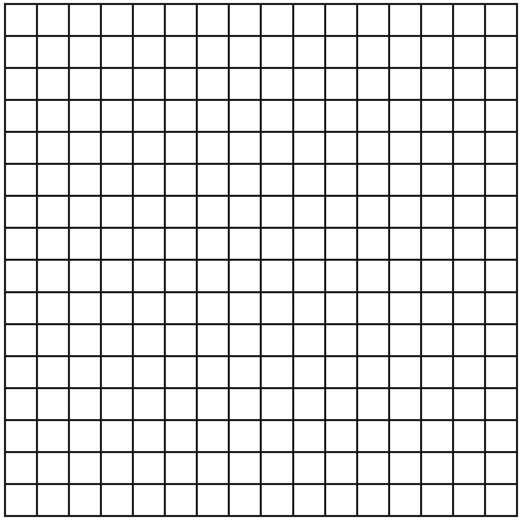
Indexing (and storage) also gets very expensive

- exponential growth in the number of dimensions

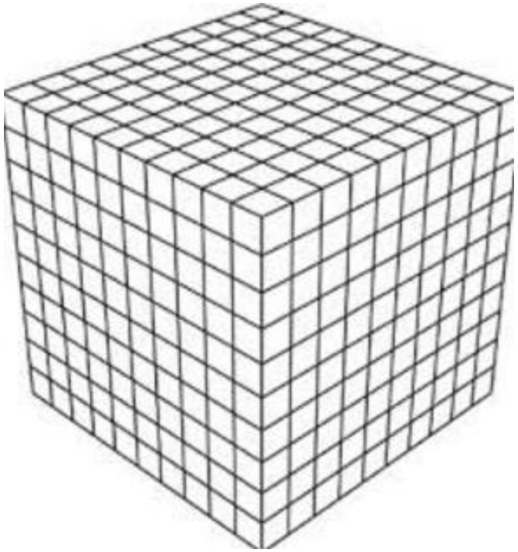


16 cells





16^2 = 256 cells



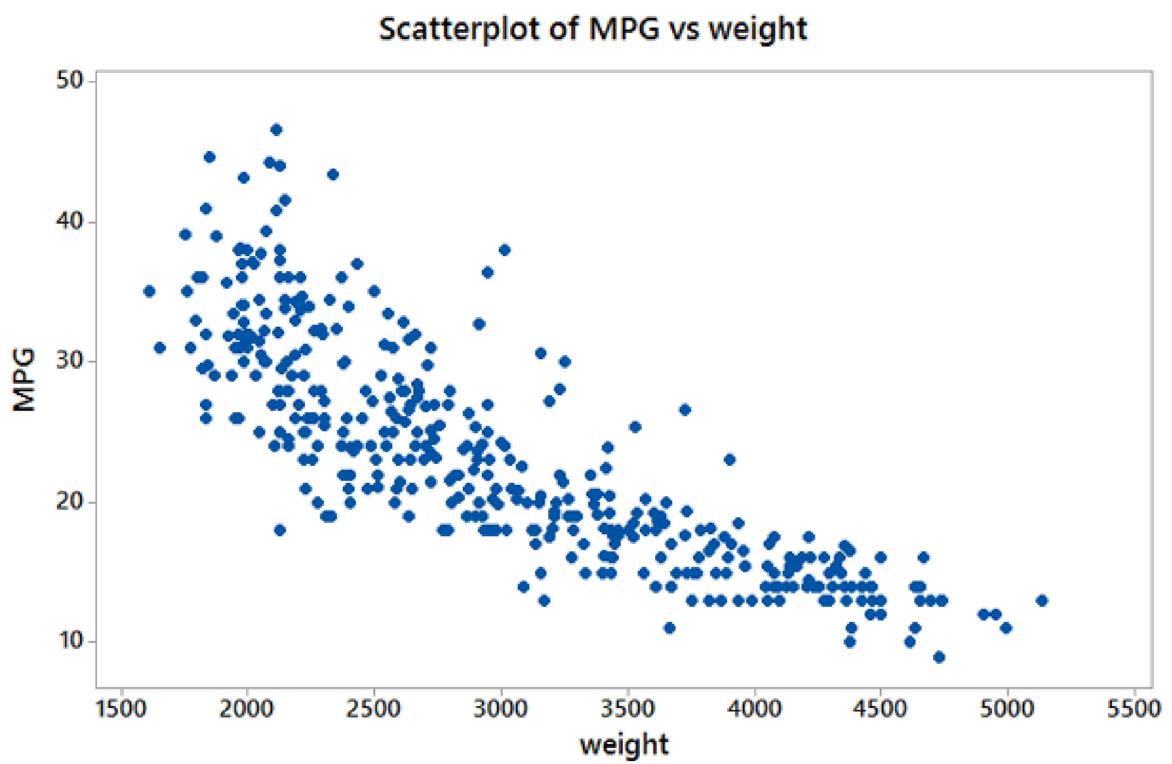
16^3 = 4,096 cells

- 4D: 65k cells   5D: 1M cells   6D: 16M cells   7D: 268M cells
- keep a keen eye on storage complexity

Scatterplot Matrix

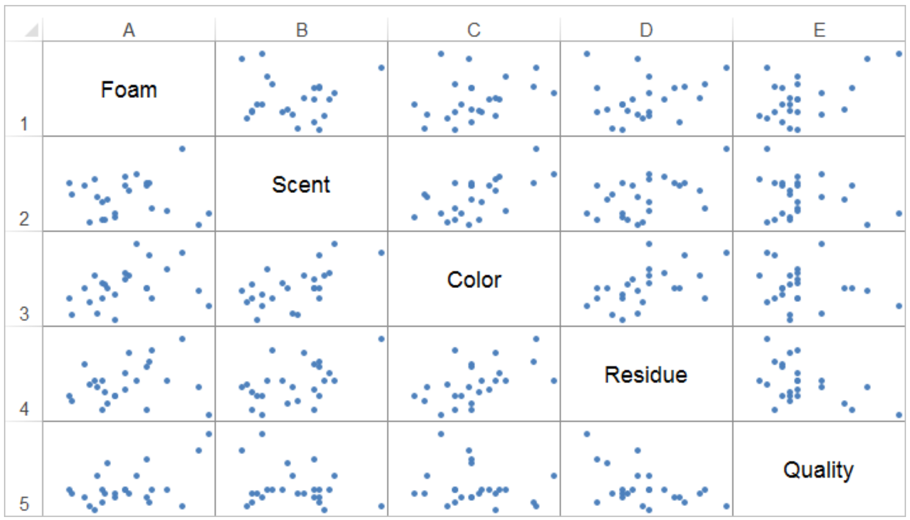
Scatterplots

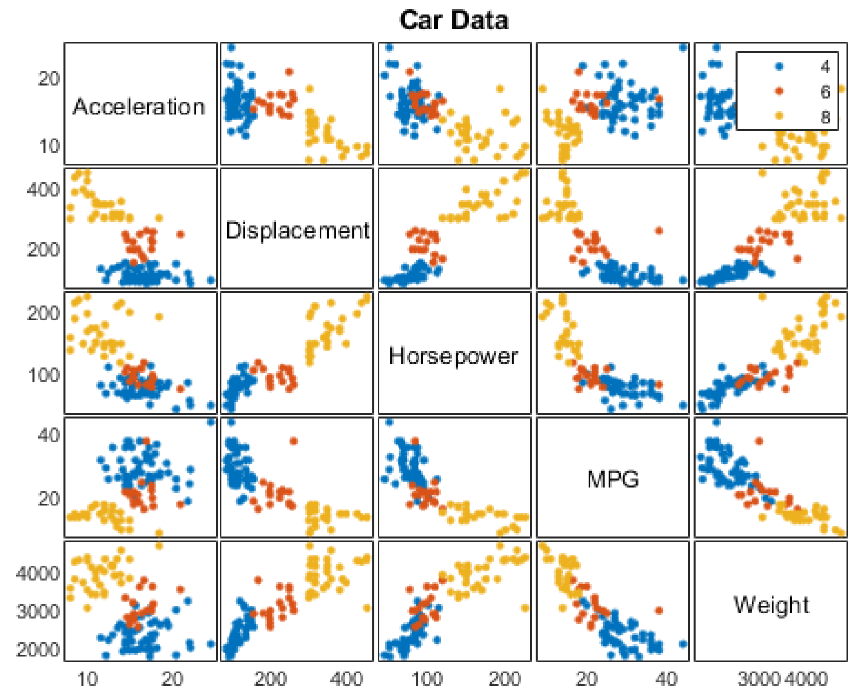
Projection of the data items into a bivariate basis of axes



But what if you have more than two variables?

Scatterplot Matrix





raw data

colored by cluster or class

Problem:

- multivariate relationships are scattered across the tiles
- difficult to see multivariate relationships
- biplots are one way to visualize these – there are others

BIPLOTS

Projection Operations

How does 2D projection work in practice?

- N-dimensional point $x = \{x_1. x_2, x_3, \ldots x_N\}$
- a basis of two orthogonal axis vectors defined in N-D space

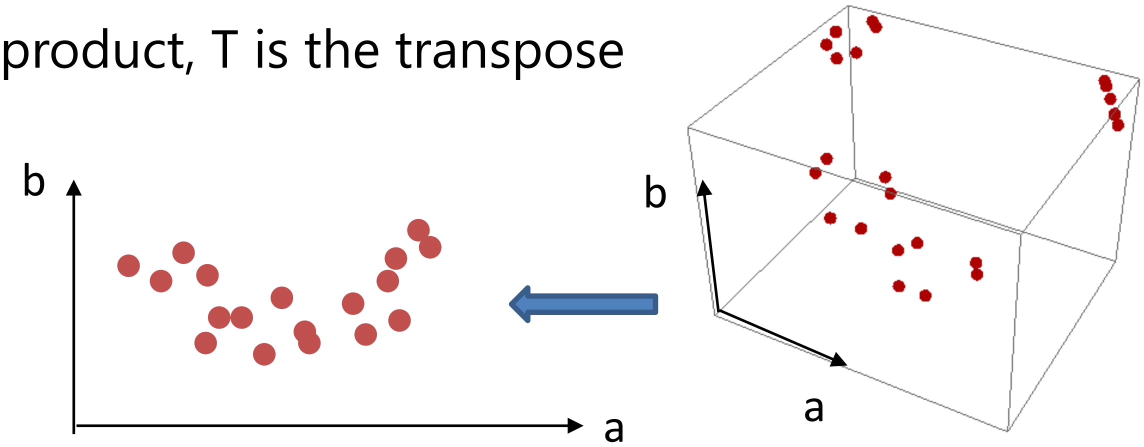
  $a = \{a_1. a_2, a_3, \ldots a_N\}$

  $b = \{b_1. b_2, b_3, \ldots b_N\}$
- a projection $\{x_a, x_b\}$ of x into the 2D basis spanned by {a, b} is:

  $x_a = a \cdot x^T$

  $x_b = b \cdot x^T$

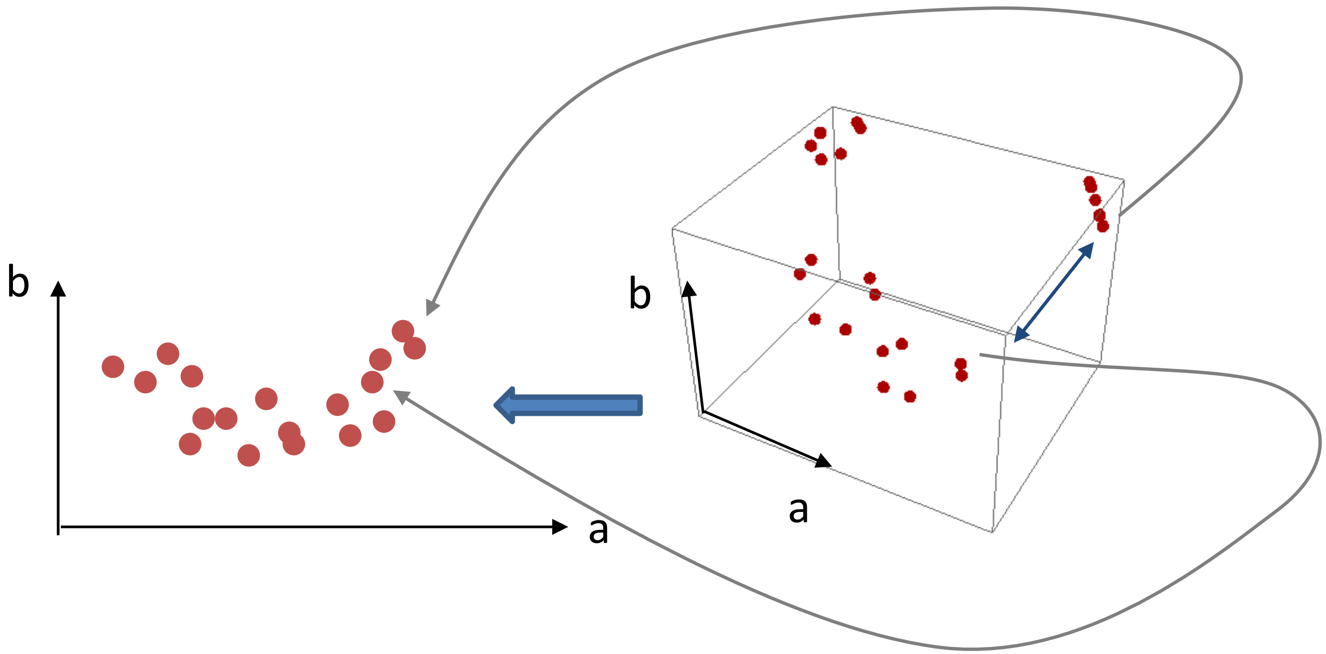
  where · is the dot product, T is the transpose



Projection Ambiguity

Projection causes inaccuracies

- close neighbors in the projections may not be close neighbors in the original higher-dimensional space
- this is called *projection ambiguity*



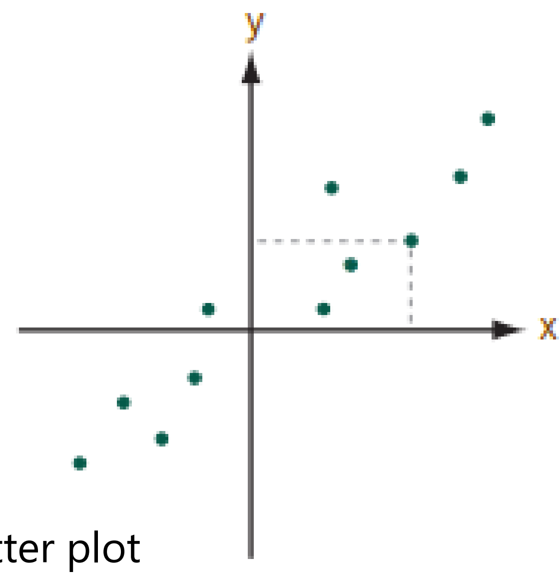
BIPLOTS

Plots data points and dimension axes into a single visualization

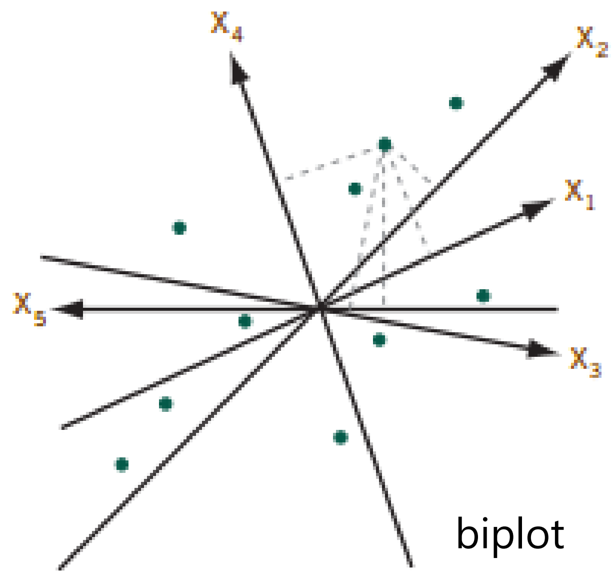
- uses first two PCA vectors as the basis to project into
- find plot coordinates [x] [y]

  for data points: [PCA_1 · data vector] [PCA_2 · data vector]

  for dimension axes: [PCA_1[dimension]] [PCA_2[dimension]]



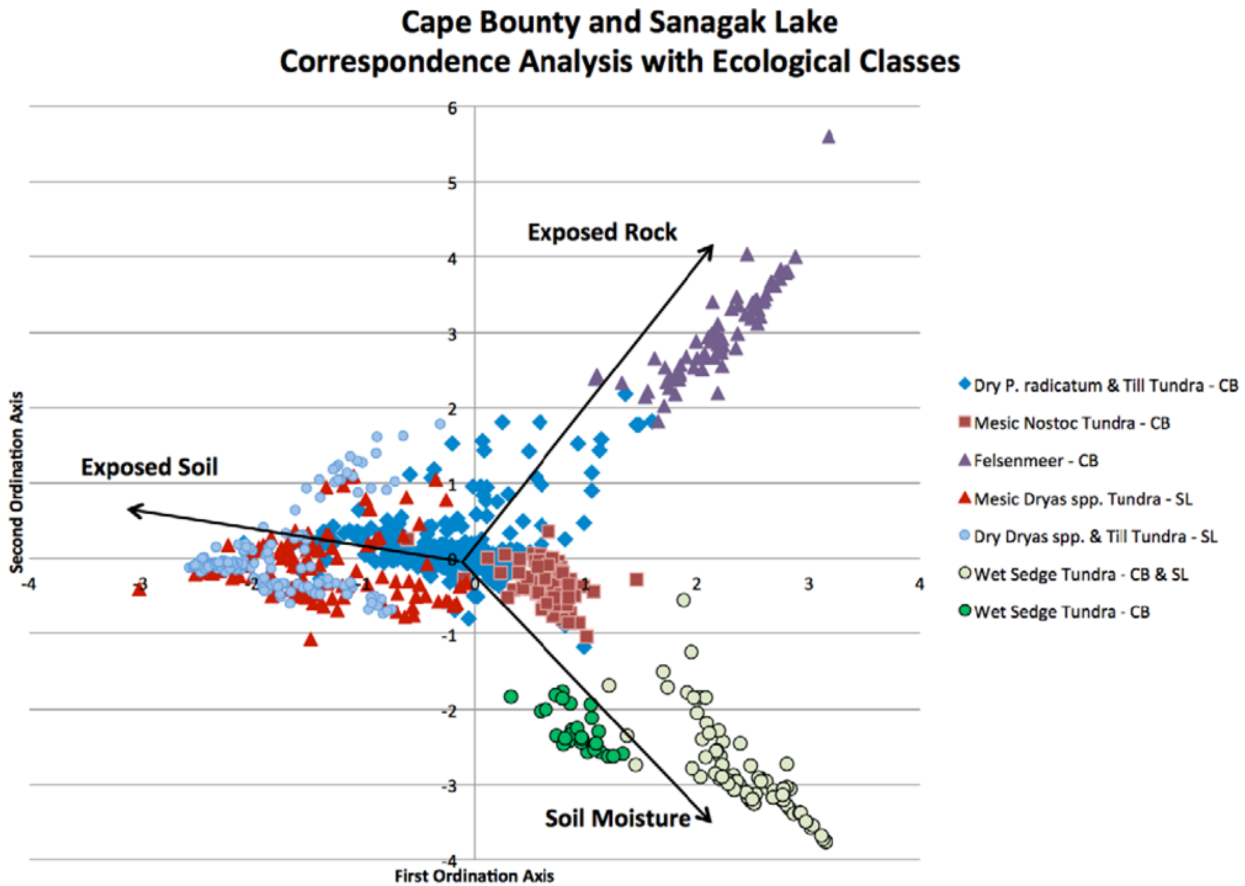
scatter plot



biplot

Biplots Can Have Projection Ambiguities

Are just a linear projection into the 2D basis generated by PCA



Biplots – A Word of caution

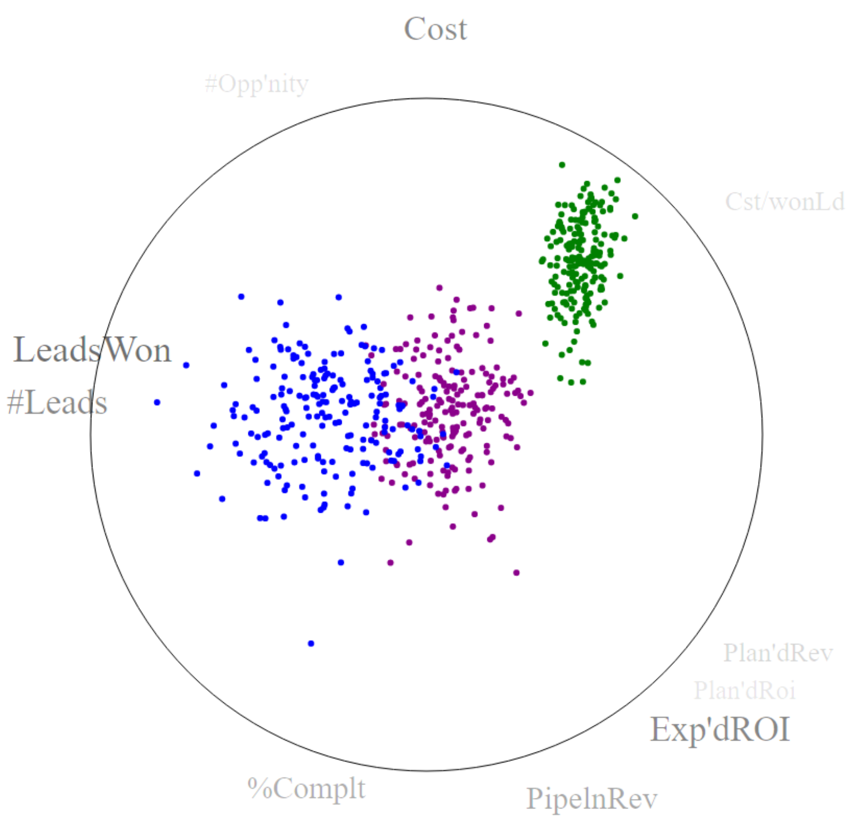
Do be aware that the projections may not be fully accurate

- you are projecting N-D into 2D by a linear transformation
- if there are more than 2 significant PCA vectors then some variability will be lost and won't be visualized
- remote data points might project into nearby plot locations suggesting false relationships → projection ambiguity
- always check out the PCA scree plot to gauge accuracy

Interactive Biplots

Also called multivariate scatterplot

- biplot-axes length vis replaced by graphical design
- less cluttered view
- but there's more to this .....



B. Wang, K. Mueller, "The Subspace Voyager: Exploring High-Dimensional Data along a Continuum of Salient 3D Subspaces," *IEEE TVCG, 2018*

Meet the *Subspace Voyager*

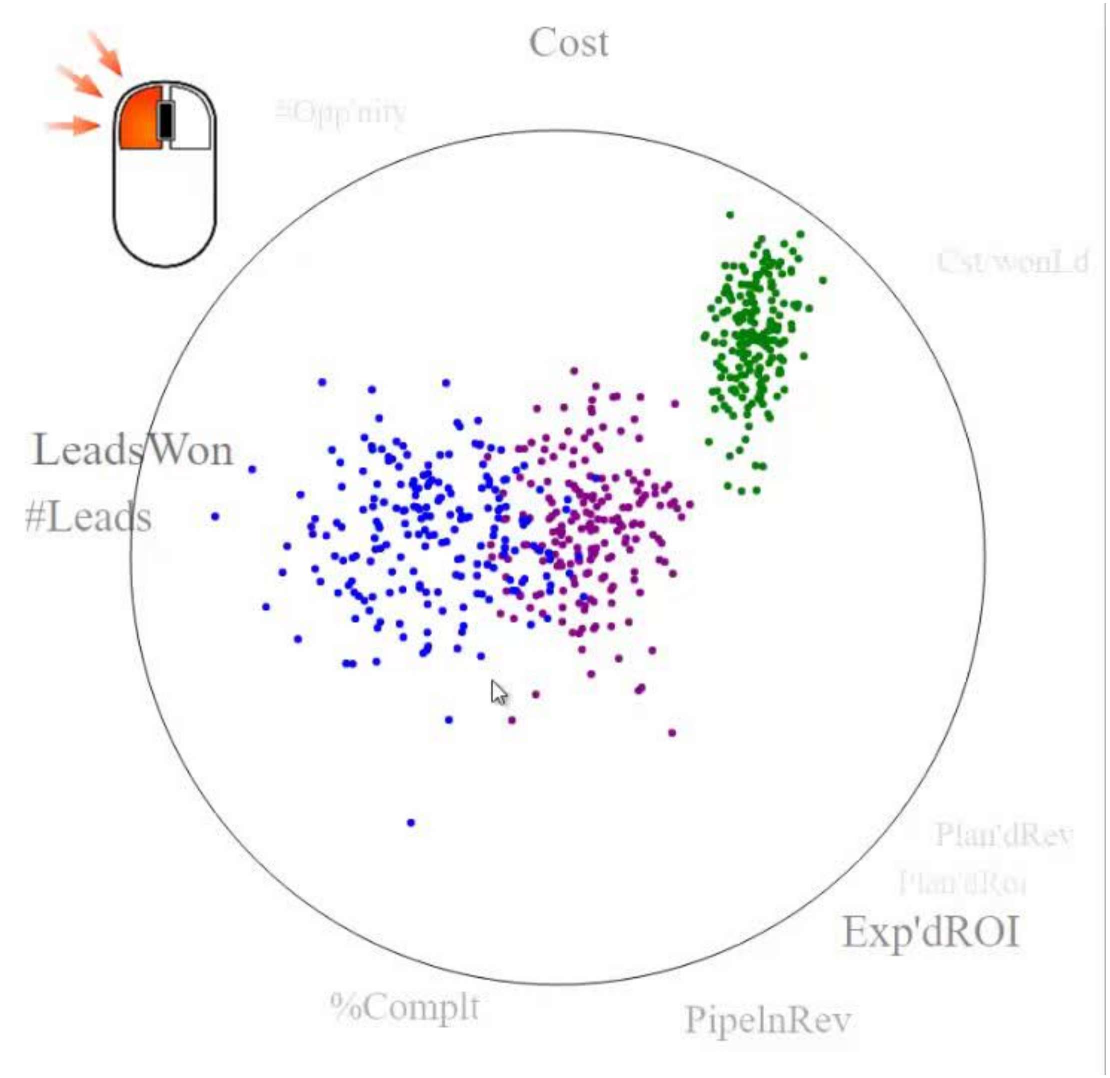
Decomposes high-D data spaces into lower-D subspaces by

- clustering
- classification
- reducing clusters to intrinsic dimensionality via local PCA

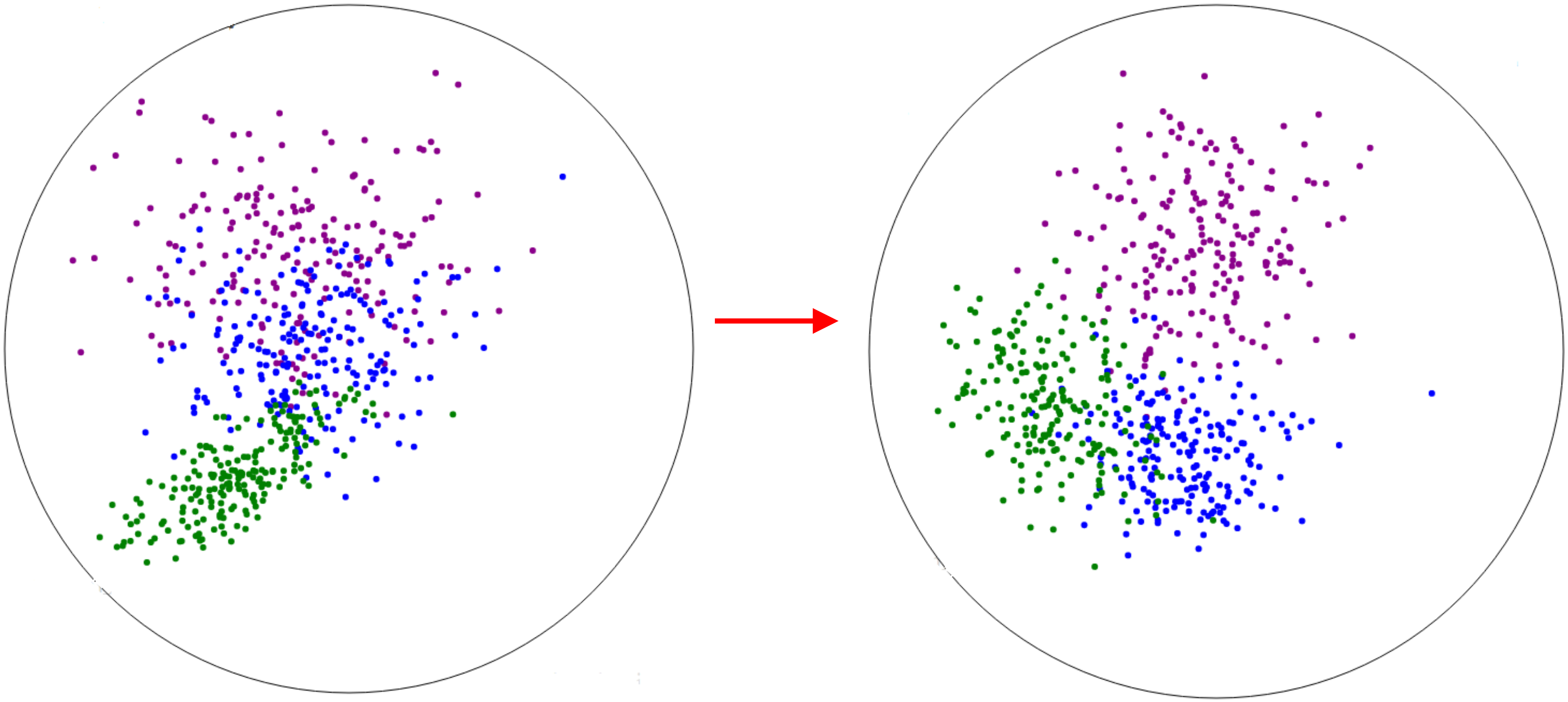
Allows users to interactively explore these lower-D subspaces

- explore them as a chain of 3D subspaces
- transition seamlessly to adjacent 3D subspaces on demand
- save observations as you go (and return to them just as well)

Trackball-Based Cluster Exploration



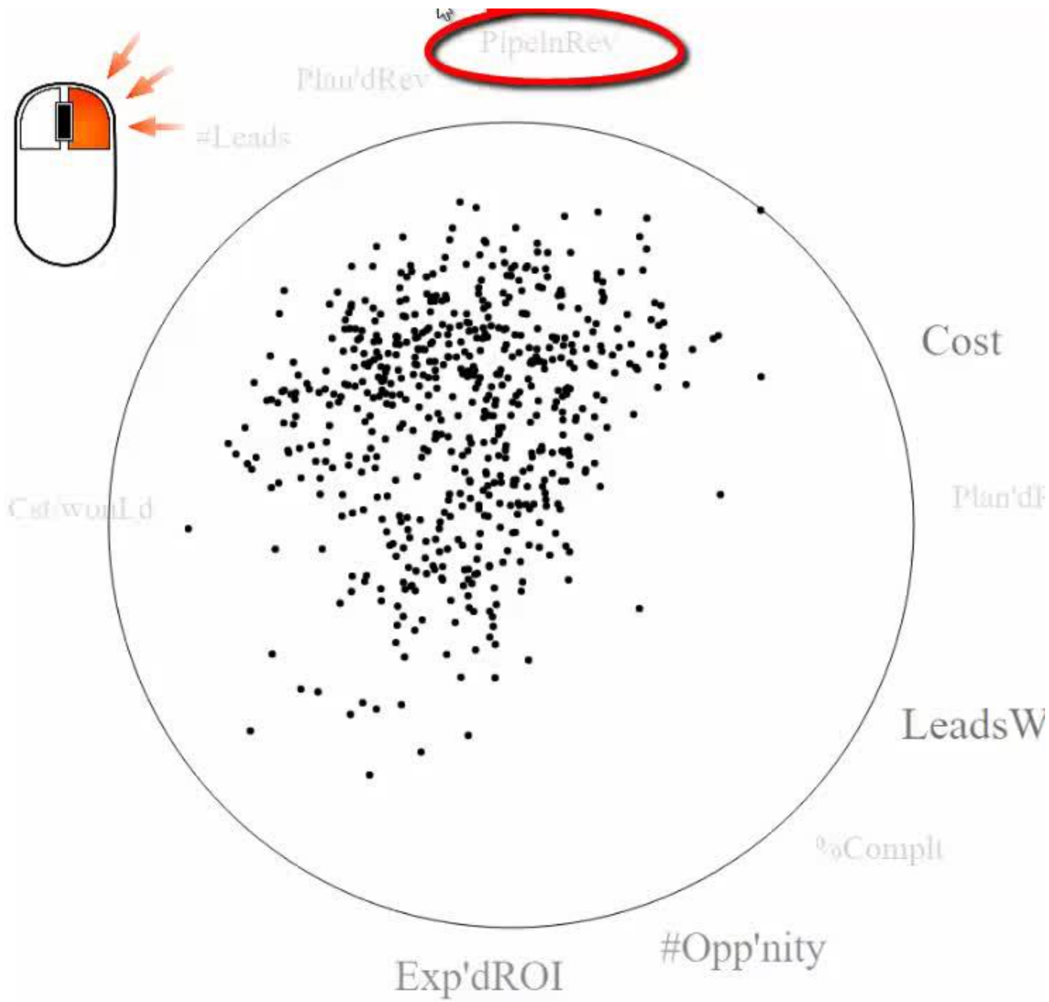
Interactive View Optimizer



Uses genetic-algorithm driven projection pursuit

Several view quality metrics are available

Chase Interesting Clusters – Transition to Adjacent 3D Subspaces



Multidimensional Scaling (MDS)

Multidimensional Scaling (MDS)

MDS preserves similarity relationships, prevents ambiguity

- scattered points in high-dimensions (N-D)
- adjacency matrices

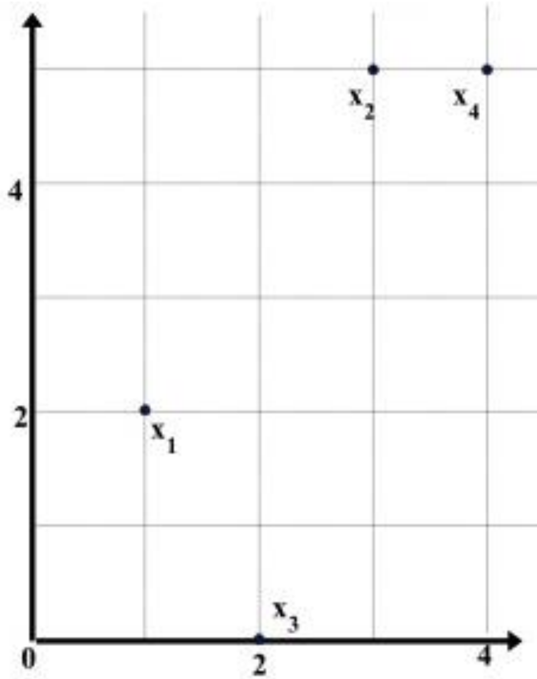
Maps the distances between observations from N-D into low-D (say 2D)

- attempts to ensure that differences between pairs of points in this reduced space match as closely as possible

The input to MDS is a distance (similarity) matrix

- actually, you use the *dissimilarity* matrix because you want similar points mapped closely
- dissimilar point pairs will have greater values and map father apart

The Dissimilarity Matrix



Data Matrix

point	attribute1	attribute2
x1	1	2
x2	3	5
x3	2	0
x4	4	5

Dissimilarity Matrix

(with Euclidean Distance)

	x1	*x2*	*x3*	*x4*
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

Distance Matrix

MDS turns a distance matrix into a network or point cloud

- correlation, cosine, Euclidian, and so on

Suppose you know a matrix of distances among cities

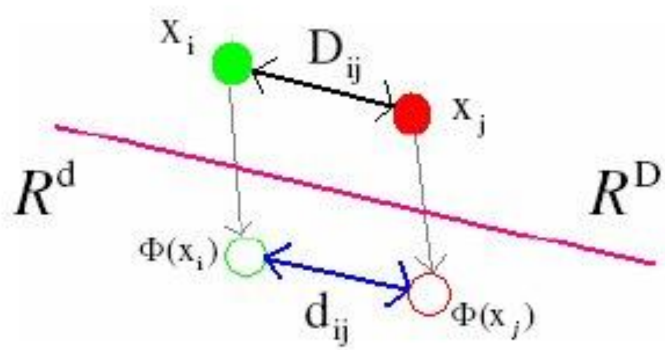
	Chicago	Raleigh	Boston	Seattle	S.F.	Austin	Orlando
Chicago	0						
Raleigh	641	0					
Boston	851	608	0				
Seattle	1733	2363	2488	0			
S.F.	1855	2406	2696	684	0		
Austin	972	1167	1691	1764	1495	0	
Orlando	994	520	1105	2565	2458	1015	0

Result of MDS

Compare with real Map

MDS Algorithm

- Task:
  - Find that configuration of image points whose pairwise distances are most similar to the original inter-point distances !!!
- Formally:
  - Define: $D_{ij} = \| x_i - x_j \|_D$ $\quad d_{ij} = \| y_i - y_j \|_d$
  - Claim: $D_{ij} \equiv d_{ij} \quad \forall i, j \in [1, n]$
- In general: an exact solution is not possible !!!
- Inter Point distances → invariance features



MDS Algorithm

<u>Strategy (of metric MDS):</u>

- iterative procedure to find a good configuration of image points
  - 1) Initialization
    → Begin with some (arbitrary) initial configuration
  - 2) Alter the image points and try to find a configuration of points that minimizes the following sum-of-squares error function:

MDS Algorithm

Strategy (of metric MDS):

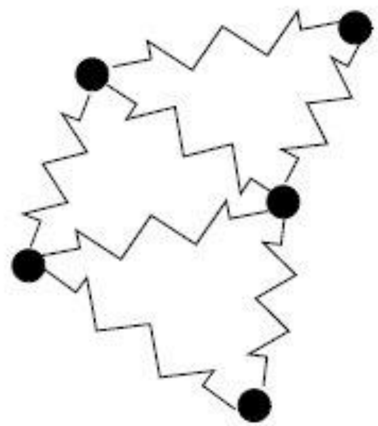
- iterative procedure to find a good configuration of image points
  - 1) Initialization
    → Begin with some (arbitrary) initial configuration
  - 2) Alter the image points and try to find a configuration of points that minimizes the following sum-of-squares error function:

$$E = \sum_{i<j}^{N} \left(D_{ij} - d_{ij}\right)^2$$

Force-Directed Algorithm

Spring-like system

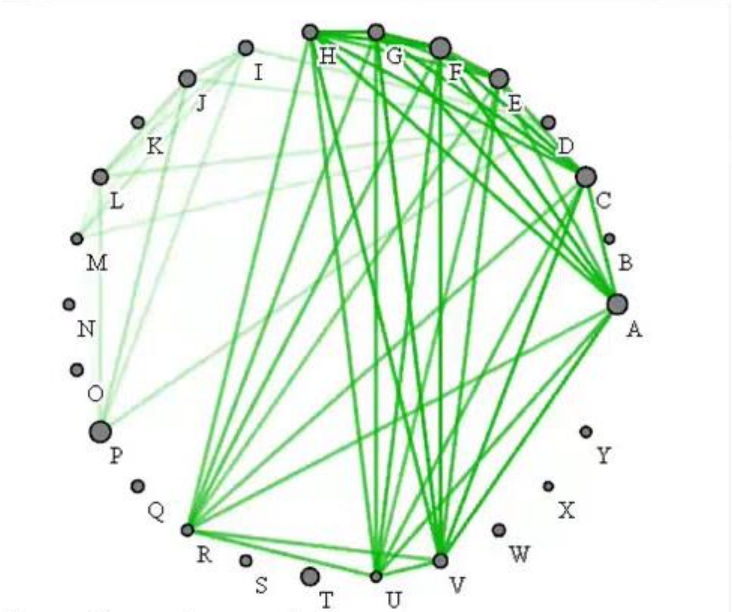
- insert springs within each node
- the length of the spring encodes the desired node distance
- start at an initial configuration
- iteratively move nodes until an energy minimum is reached



Force-Directed Algorithm

Spring-like system

- insert springs within each node
- the length of the spring encodes the desired node distance
- start at an initial configuration
- iteratively move nodes until an energy minimum is reached



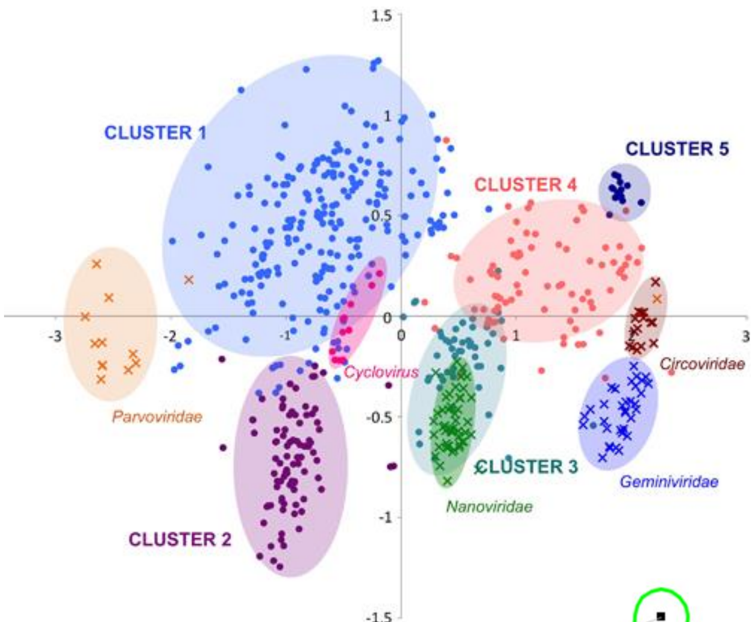
Vertex layout by correlations.

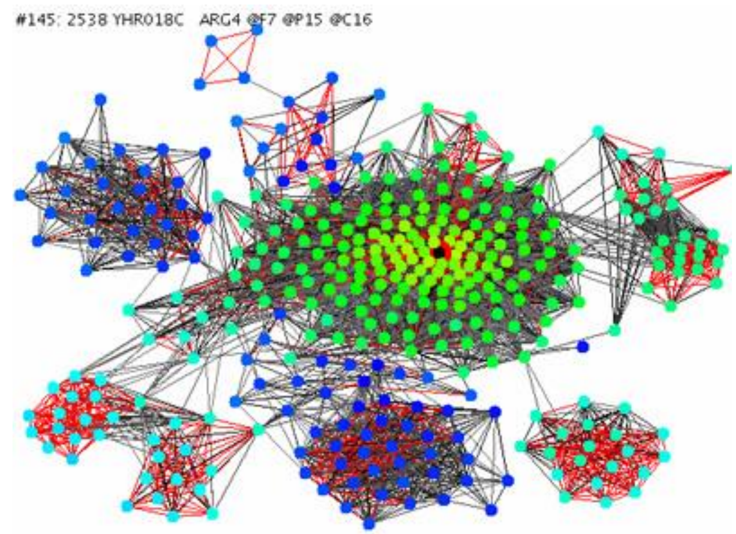
Uses of MDS

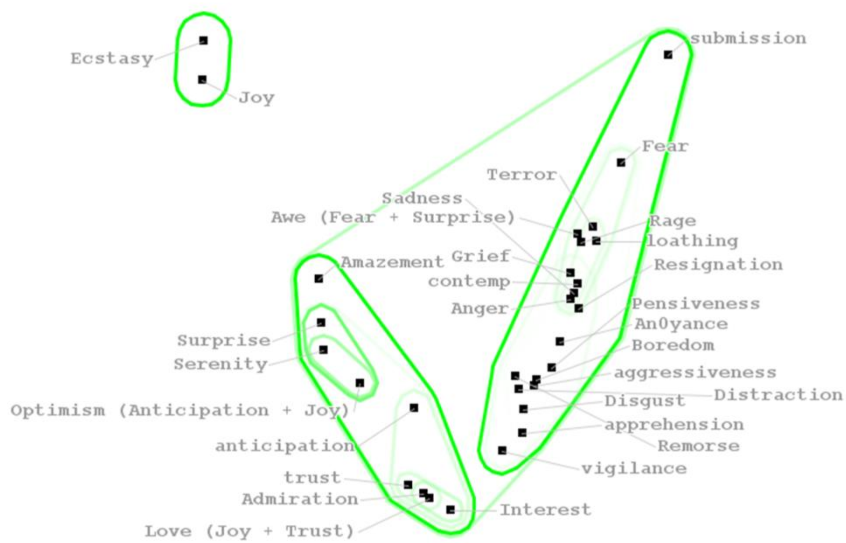
Distance (similarity) metric

- Euclidian distance (best for data)
- Cosine distance (best for data)
- 1-|correlation| distance (best for attributes)
- use | | if you do not care about positive or negative correlations
- leave off | | if you want positively correlated attribute points closer

MDS Examples







MDS in Scikit-learn

`sklearn.manifold`.MDS

class `sklearn.manifold.MDS`(*n_components=2, metric=True, n_init=4, max_iter=300, verbose=0, eps=0.001, n_jobs=1, random_state=None, dissimilarity='euclidean'*) [source]

sklearn.manifold.**MDS**(

n_components=2,

metric=True,

n_init=4, Number of time the smacof algorithm will be run with different initialisation.
The final results will be the best output of the n_init consecutive runs in terms of stress.

max_iter=300, Maximum number of iterations of the SMACOF algorithm for a single run

verbose=0,

eps=0.001, relative tolerance w.r.t stress to declare converge

n_jobs=1,

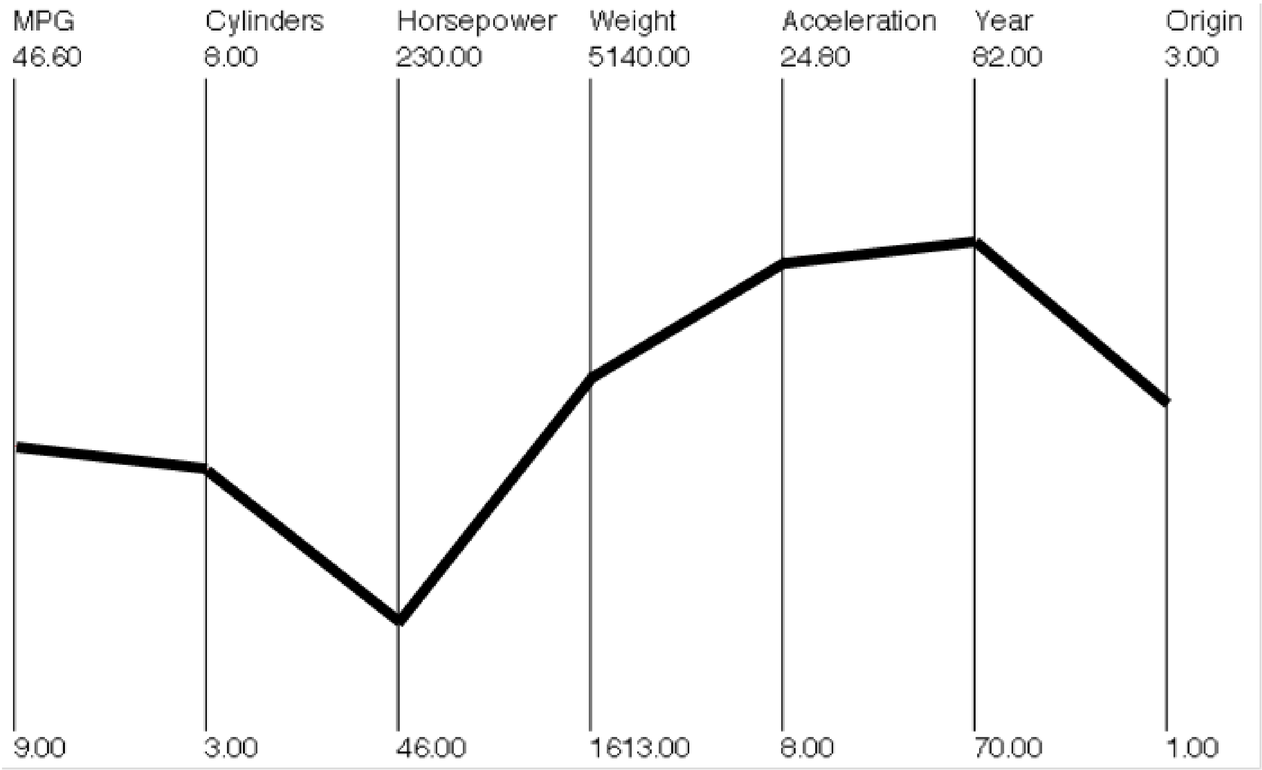
random_state=None,

dissimilarity='euclidean') Which dissimilarity measure to use. Supported are 'euclidean' and 'precomputed'.

The **SMACOF** (Scaling by MAjorizing a COmplicated Function) algorithm is a multidimensional scaling algorithm which minimizes an objective function (the *stress*) using a majorization technique.

Parallel Coordinates

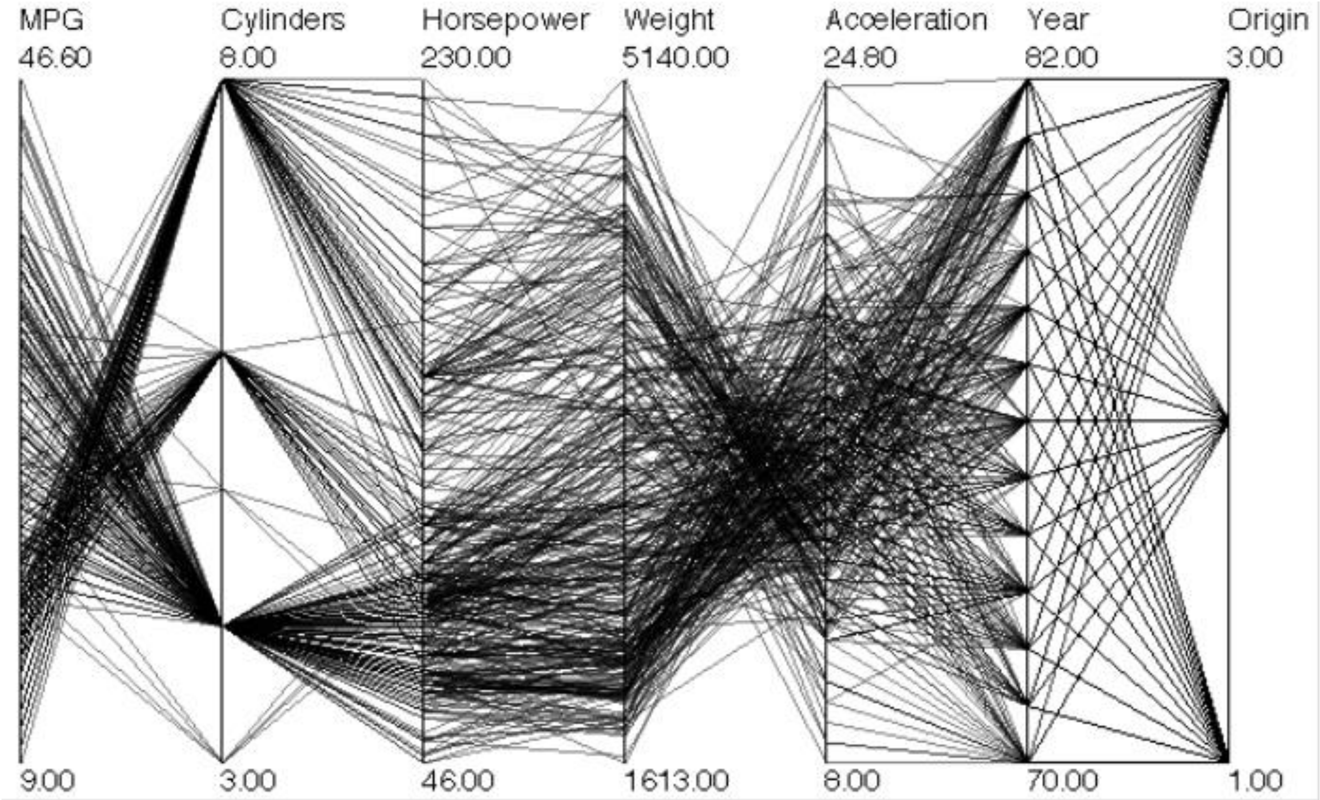
Parallel Coordinates – 1 Car



The N=7 data axes are arranged side by side

- in parallel

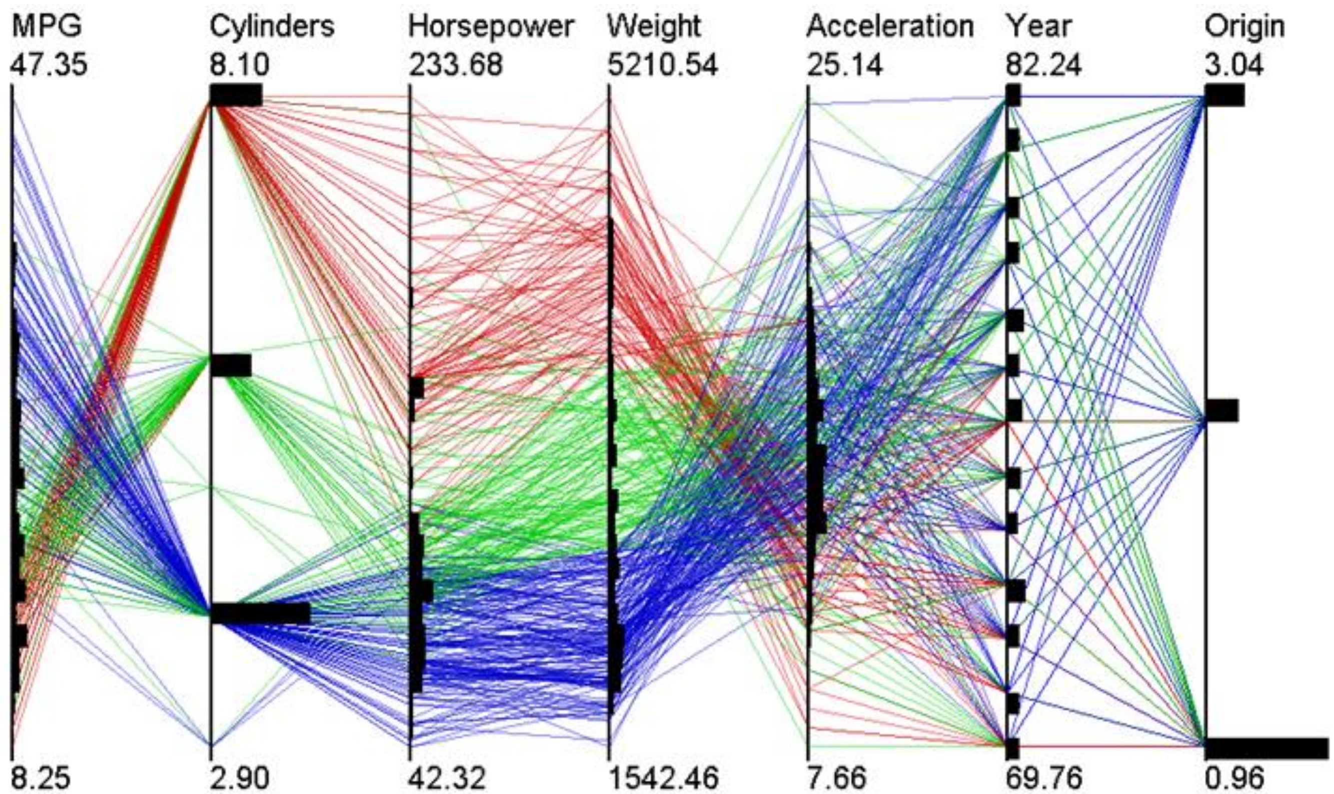
Parallel Coordinates – 100 Cars



Hard to see the individual cars?

- what can we do?

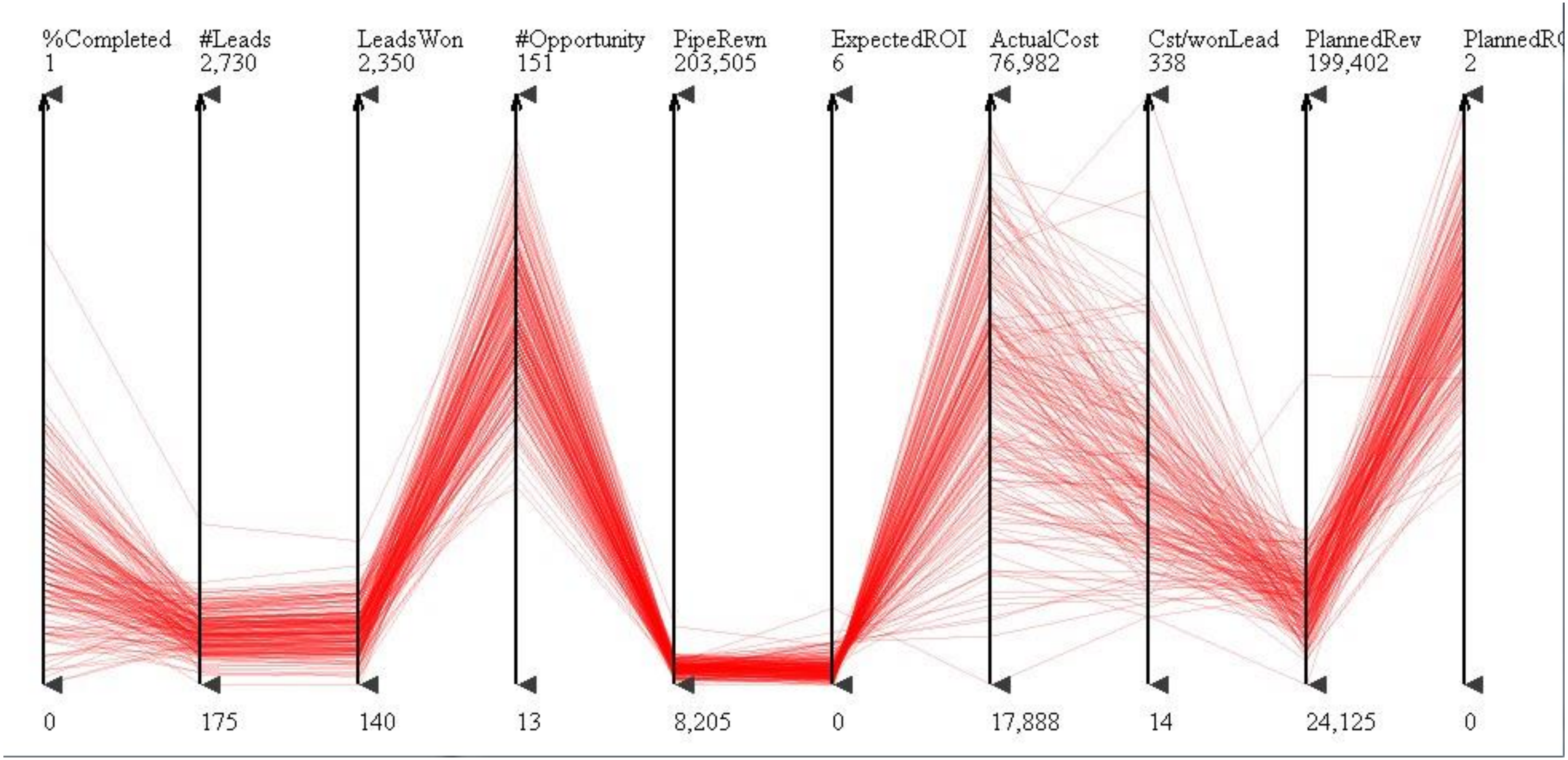
Parallel Coordinates – 100 Cars



Grouping the cars into sub-populations

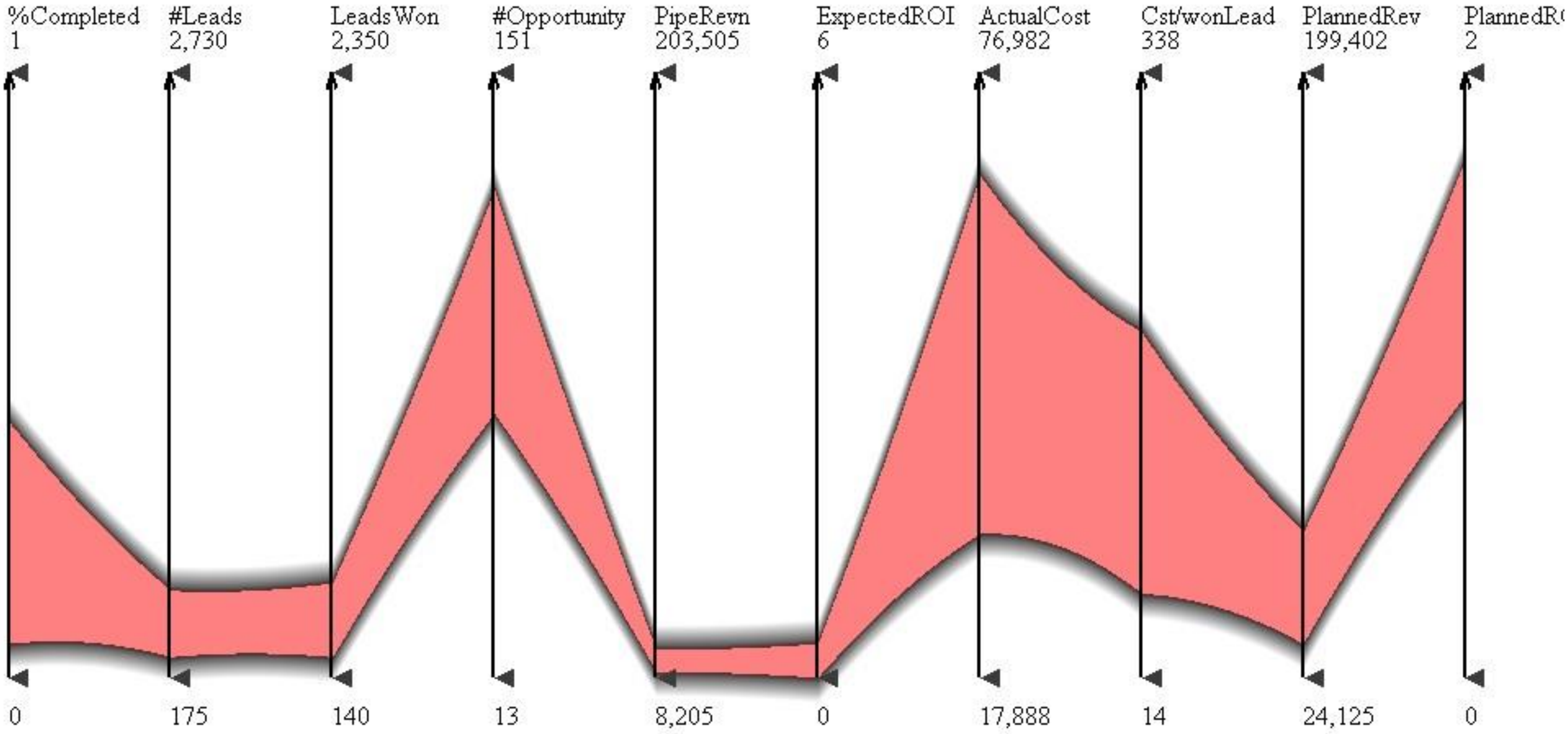
- we perform clustering
- an be automated or interactive (put the user in charge)

PC With Illustrative Abstraction



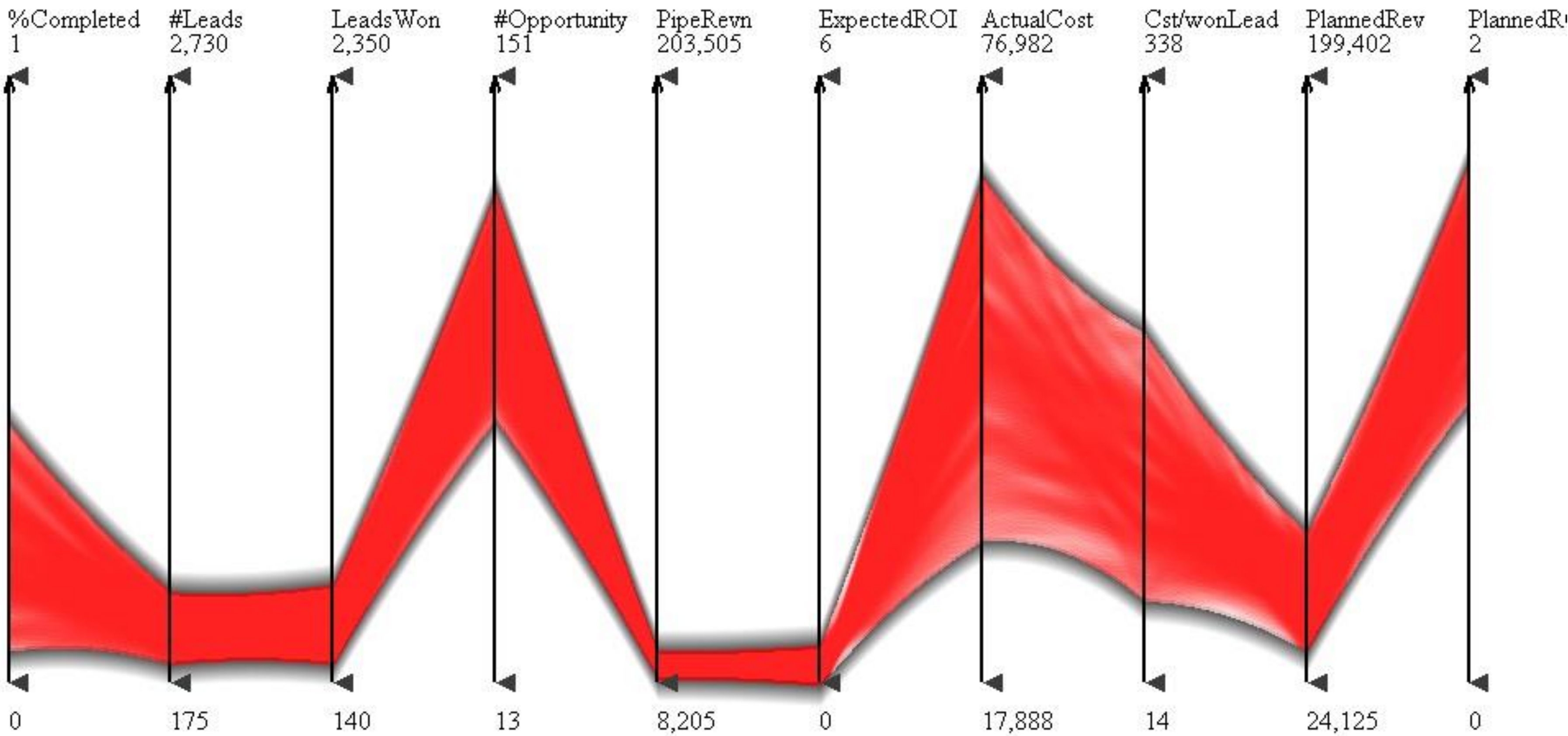
individual polylines

PC With Illustrative Abstraction



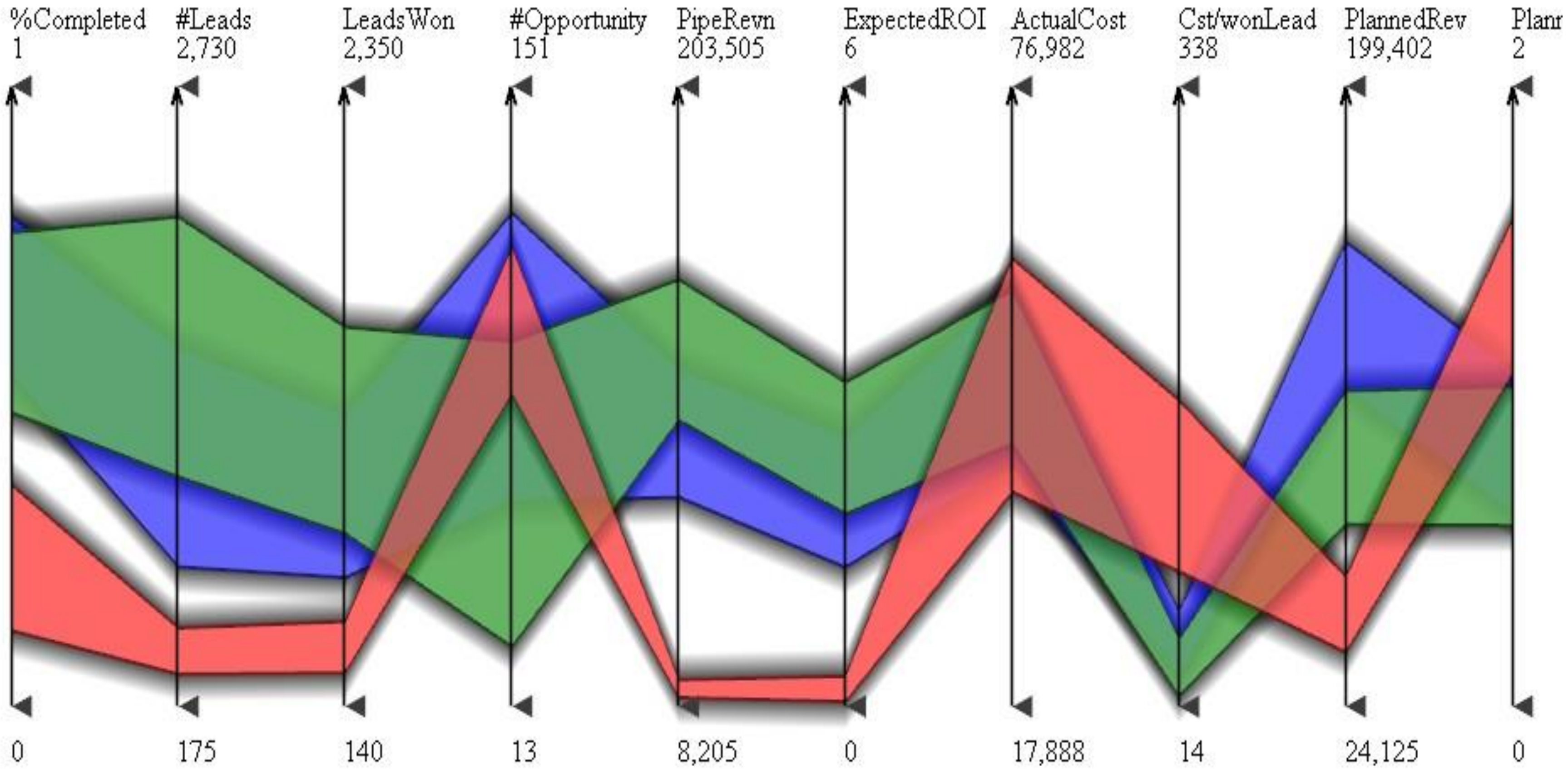
completely abstracted away

PC With Illustrative Abstraction



blended partially

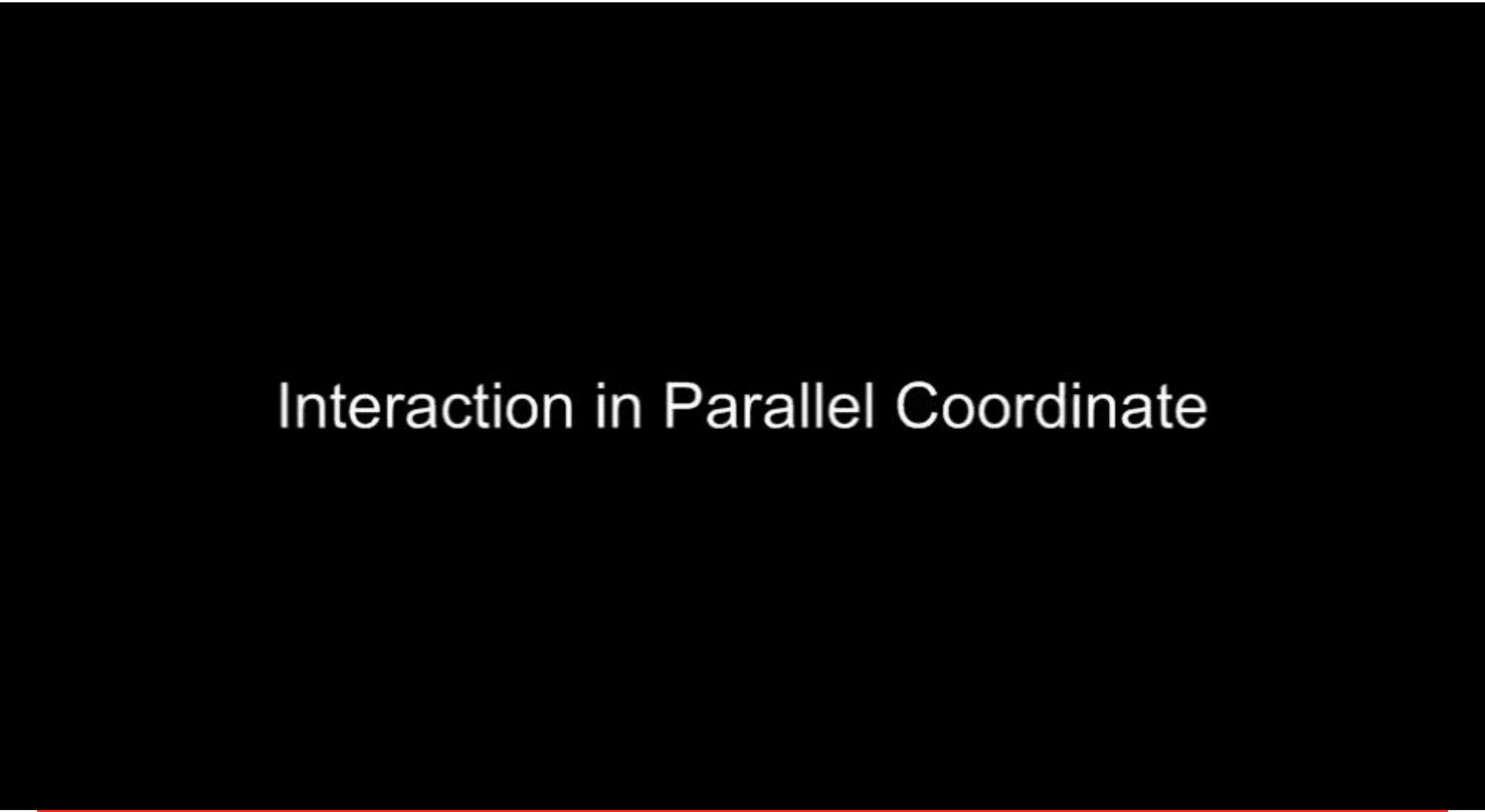
PC With Illustrative Abstraction



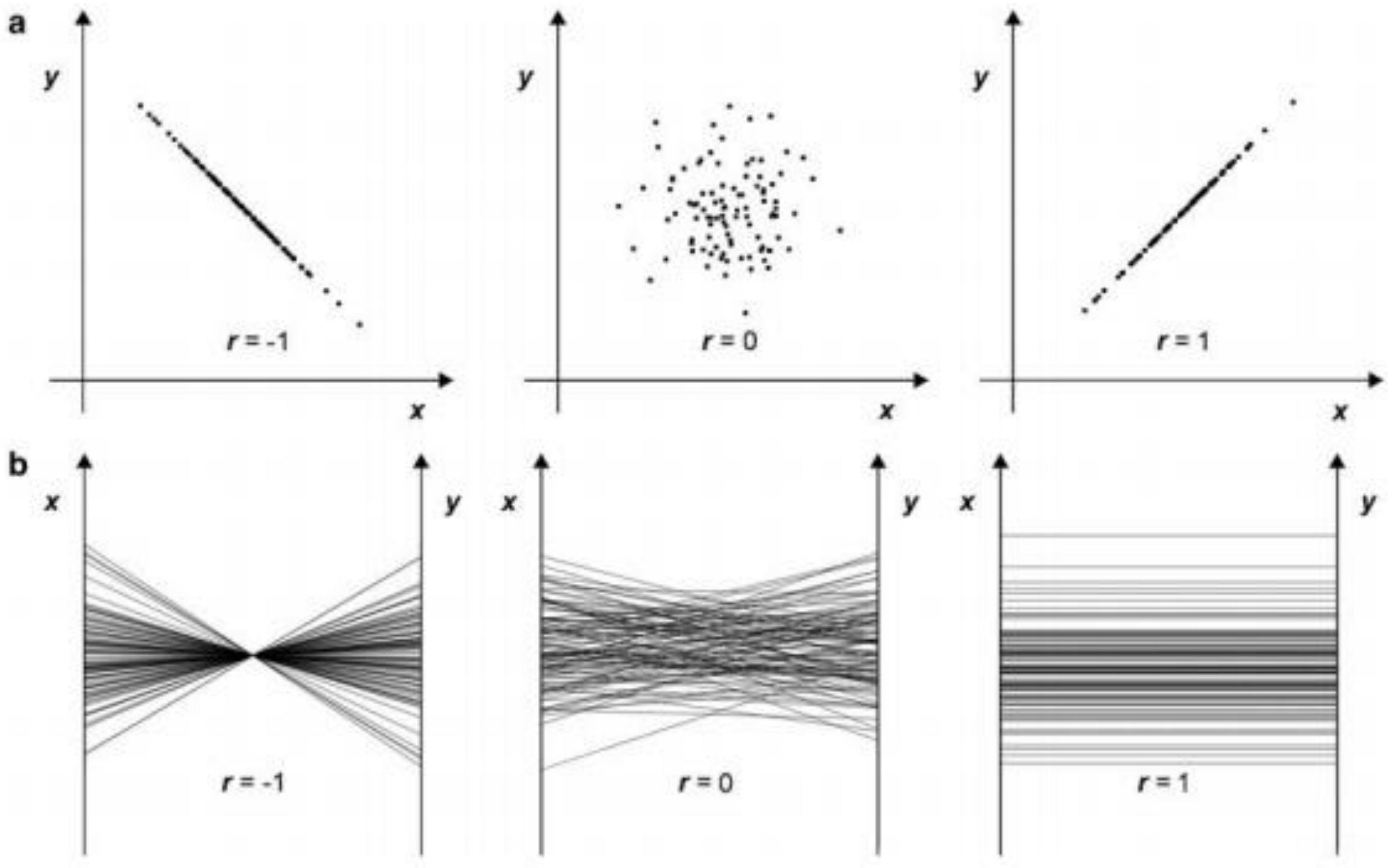
all put together – three clusters

[McDonnell and Mueller, 2008]

Interaction is Key

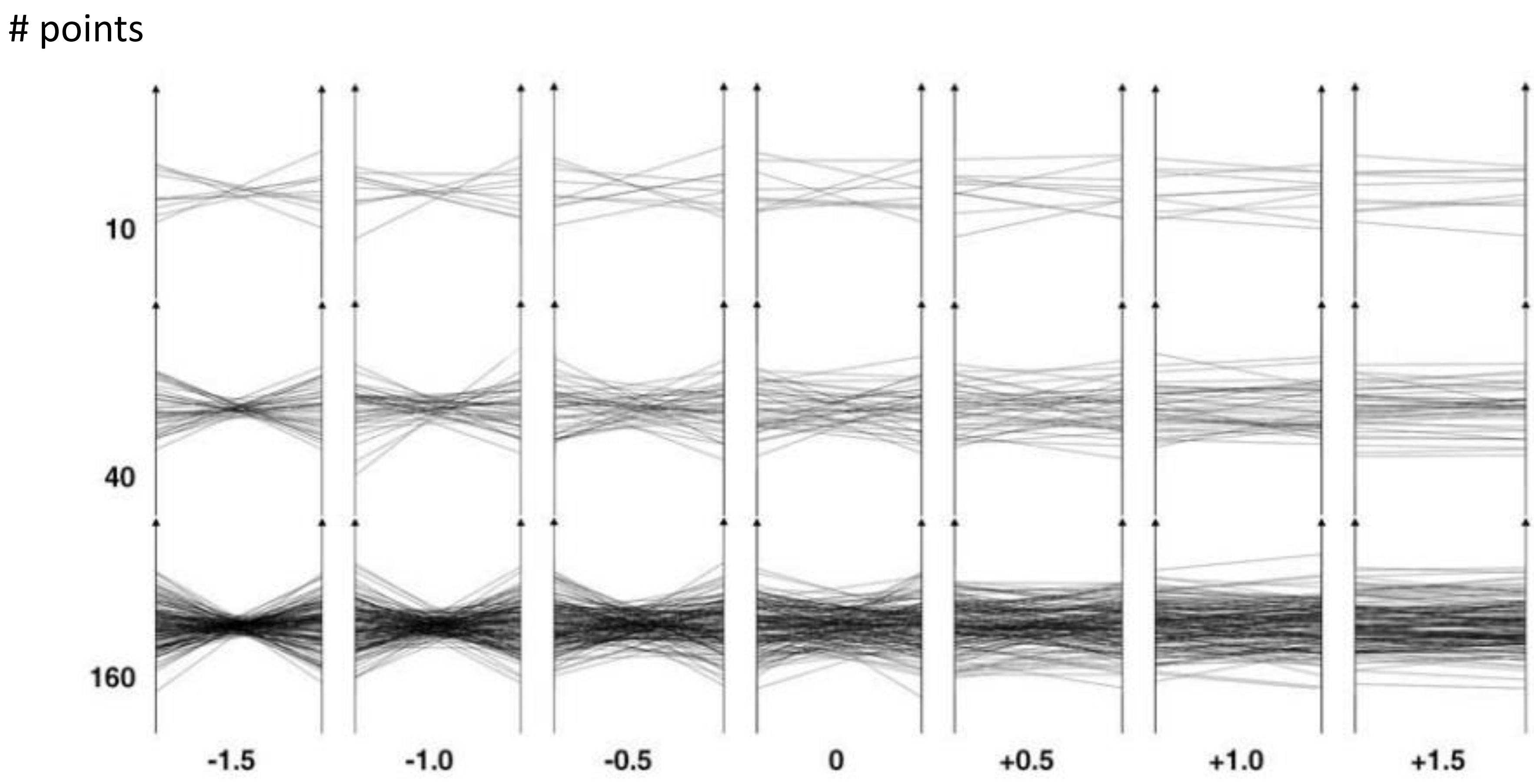


Patterns in Parallel Coordinates



correlation r=-1.0 r=0 r=1.0

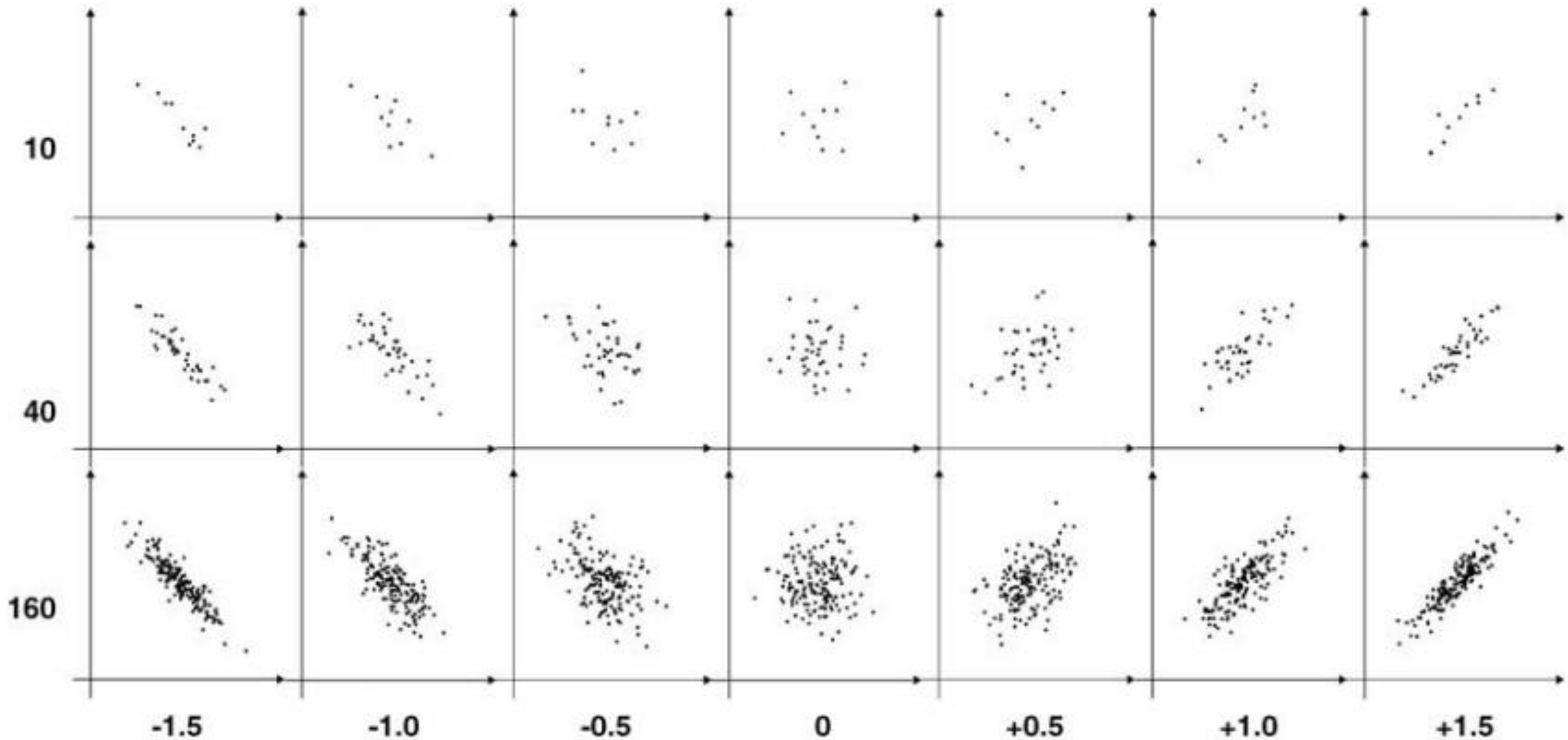
Patterns in Parallel Coordinates



Fisher-z (corresponding to $\rho = 0, \pm 0.462, \pm 0.762, \pm 0.905$)

Patterns in Scatterplots

points



Fisher-z (corresponding to $\rho = 0, \pm 0.462, \pm 0.762, \pm 0.905$)

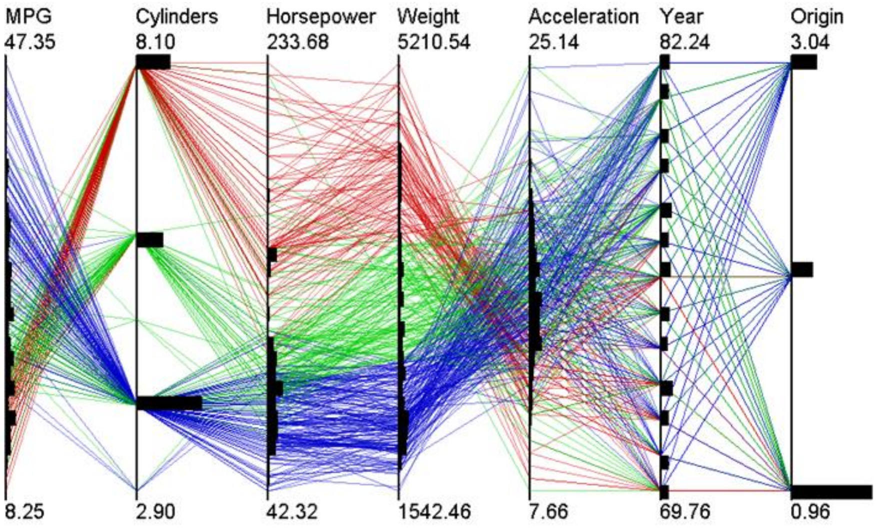
Li et al. found that <u>twice as many</u> correlation levels can be distinguished with scatterplots

Information Visualization Vol. 9, 1, 13 – 30

Axis Reordering Problem

There are n! ways to order the n dimensions

- how many orderings for 7 dimensions?
- 5,040
- but since can see relationships across 3 axes a better estimate is n!/((n-3)! 3!) = 35
- still a lot of axes orderings to try out → we need help

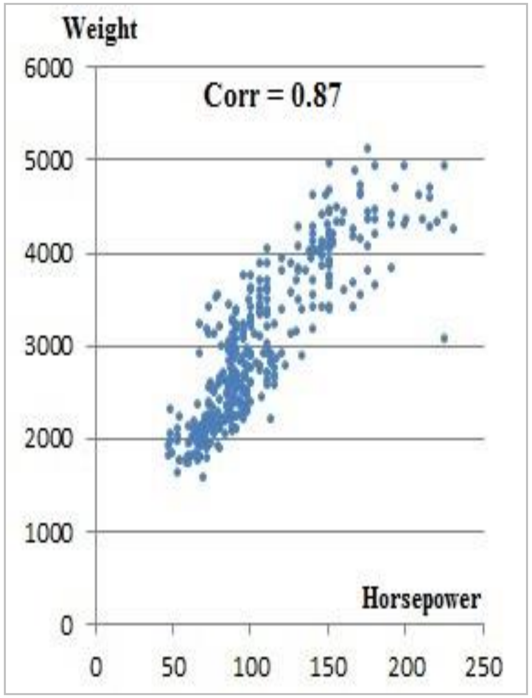


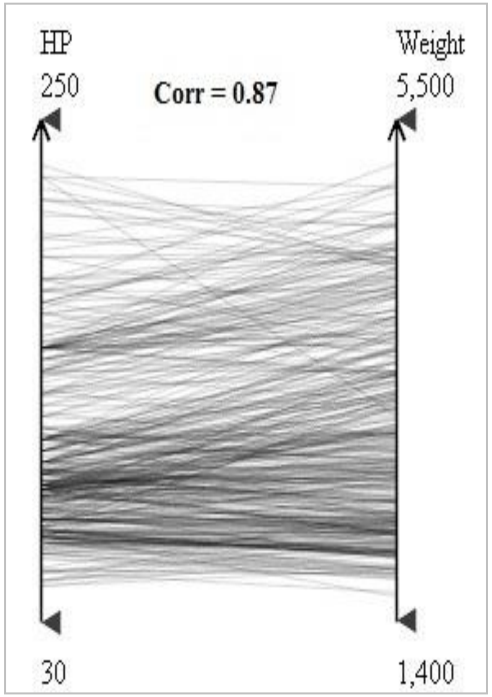
We Need a Measure for Relationships

Correlation

- a statistical measure that indicates the extent to which two or more variables fluctuate together

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$





Building the Correlation Matrix

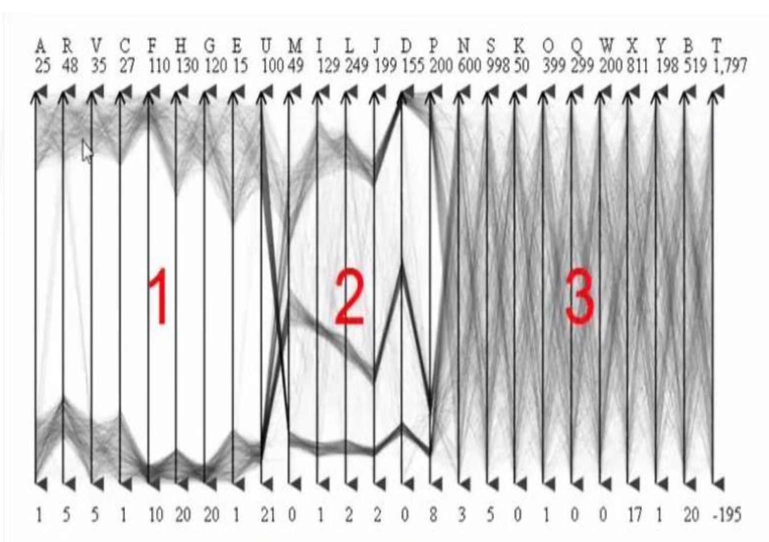
Z. Zhang, K. McDonnell, K. Mueller, "A Network-Based Interface for the Exploration of High-Dimensional Data Spaces, " *IEEE Pacific Vis*, 2012

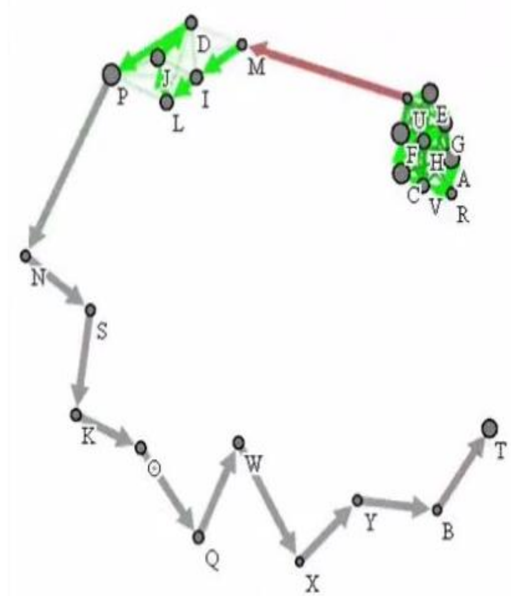
Create a correlation matrix

Run a mass-spring model

Run Traveling Salesman on the correlation nodes

Use it to order your parallel coordinate axes via TSP

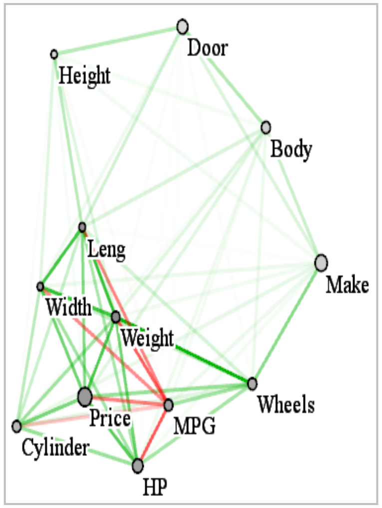




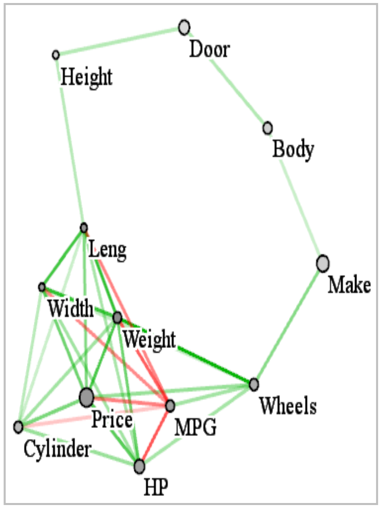
	MRK	MSFT	PFE	PG	*T*	TRV	UTX	VZ	WMT	XOM
MRK	1.	0.39	0.72	−0.43	0.57	0.031	−0.26	0.61	−0.11	−0.25
MSFT	0.39	1.	0.14	0.11	0.56	0.25	0.25	0.67	−0.074	0.24
PFE	0.72	0.14	1.	−0.77	0.08	−0.37	−0.65	0.19	−0.077	−0.72
PG	−0.43	0.11	−0.77	1.	0.25	0.68	0.92	0.086	0.072	0.9
T	0.57	0.56	0.08	0.25	1.	0.65	0.46	0.87	−0.059	0.54
TRV	0.031	0.25	−0.37	0.68	0.65	1.	0.83	0.43	−0.0067	0.81
UTX	−0.26	0.25	−0.65	0.92	0.46	0.83	1.	0.27	−0.033	0.93
VZ	0.61	0.67	0.19	0.086	0.87	0.43	0.27	1.	0.026	0.36
WMT	−0.11	−0.074	−0.077	0.072	−0.059	−0.0067	−0.033	0.026	1.	0.032
XOM	−0.25	0.24	−0.72	0.9	0.54	0.81	0.93	0.36	0.032	1.

Interaction with the Correlation Network

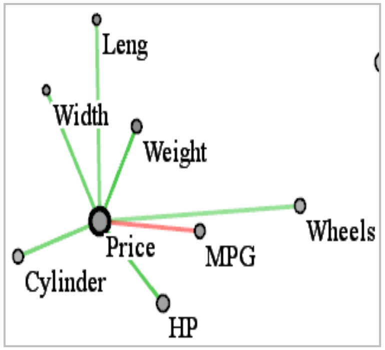
- Vertices are attributes, edges are correlations
    - vertex: size determined by $\sum_{j=0}^{D} \frac{|correlation(i,j)|}{D-1} \; j \neq i$
    - edge length is a measure of (1-|correlation|)
    - edge: color/intensity → sign/strength of correlation



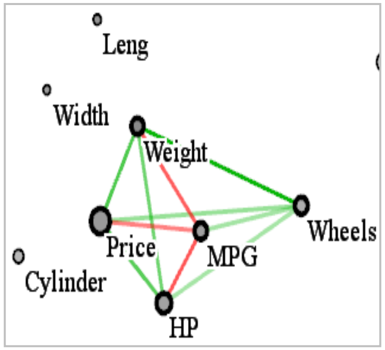
all edges



filtered by strength

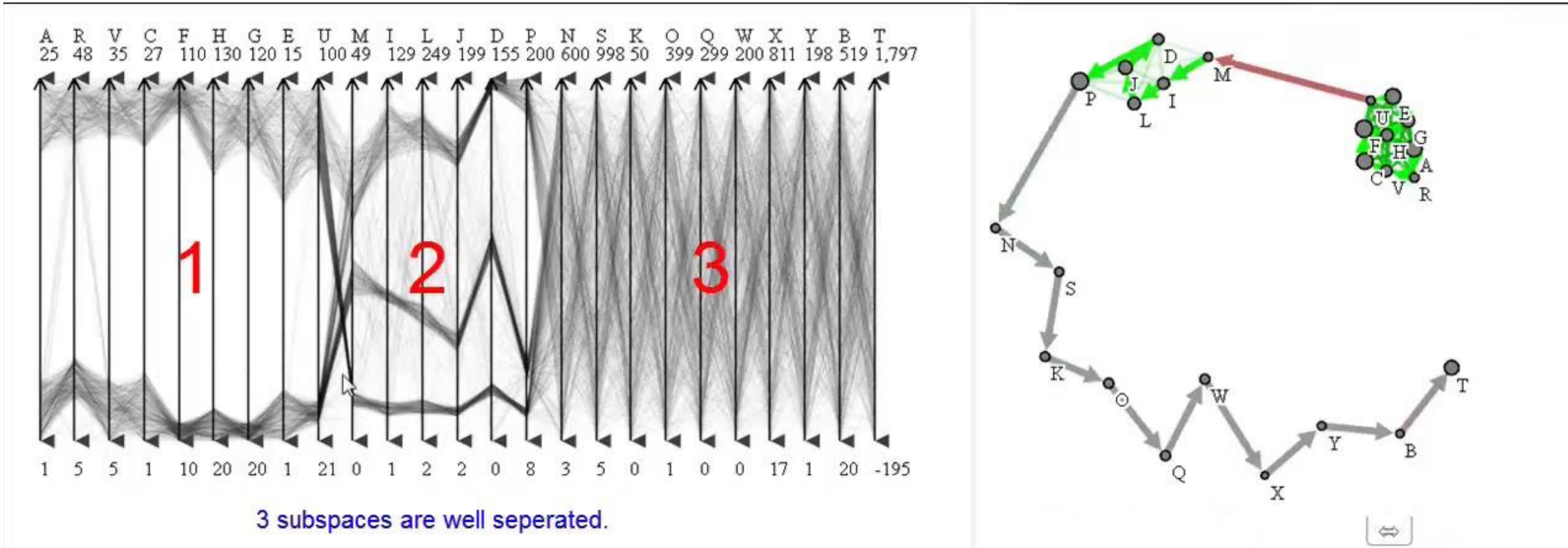


attribute centric



subset of attributes

Multiscale Zooming

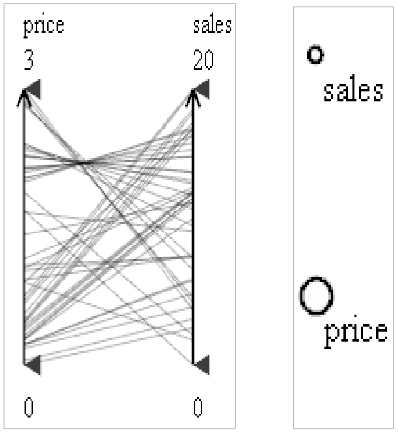


Z. Zhang, K. McDonnell, K. Mueller, "A Network-Based Interface for the Exploration of High-Dimensional Data Spaces, " *IEEE Pacific Vis*, 2012

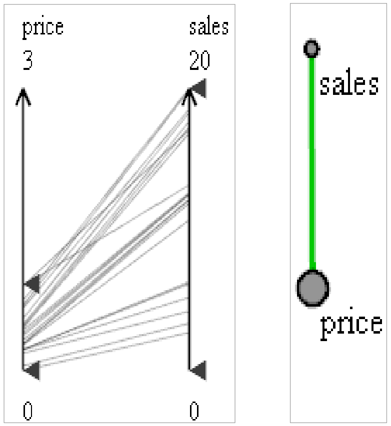
Bracketing and Conditioning

Correlation strength can often be improved by constraining a variable's value range

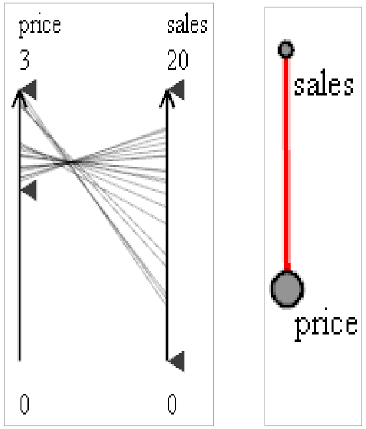
- this limits the derived relationships to this value range
- such limits are commonplace in targeted marketing, etc.



no bracketing



lower price range



higher price range

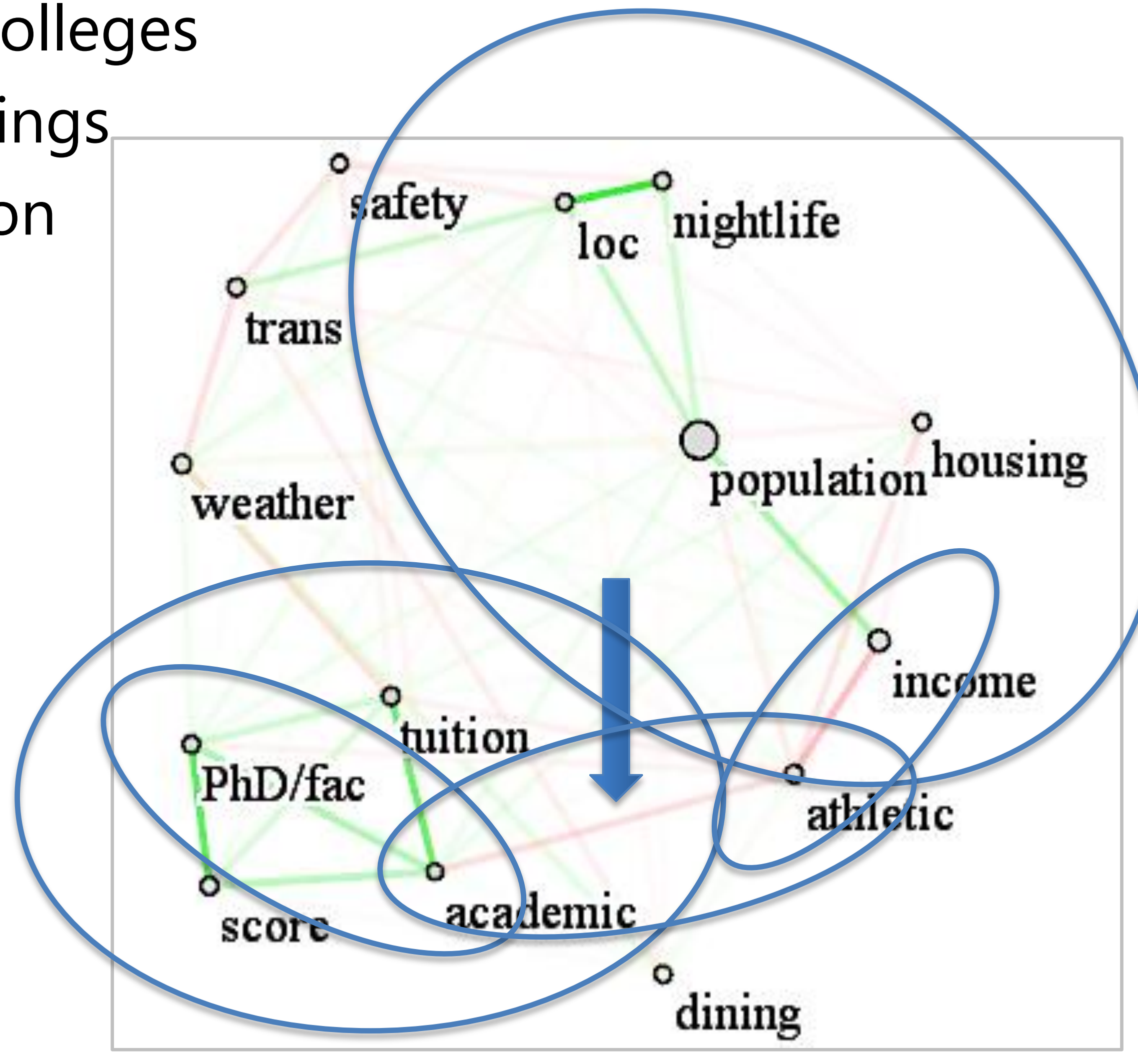
Z. Zhang, K. McDonnell, E. Zadok, K. Mueller, "Visual Correlation Analysis of Numerical and Categorical Data on the Correlation Map," *IEEE TVCG,* 2015.

Correlation Plots Are Powerful

Fused dataset of 50 US colleges

US News: academic rankings

College Prowler: survey on campus life attributes

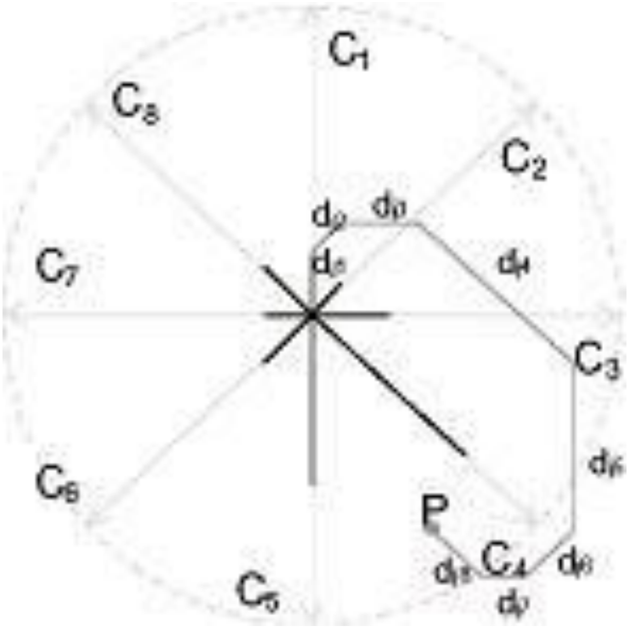


Radial Layouts

Star Coordinates

Coordinate system based on axes positioned in a “star”, or circular pattern

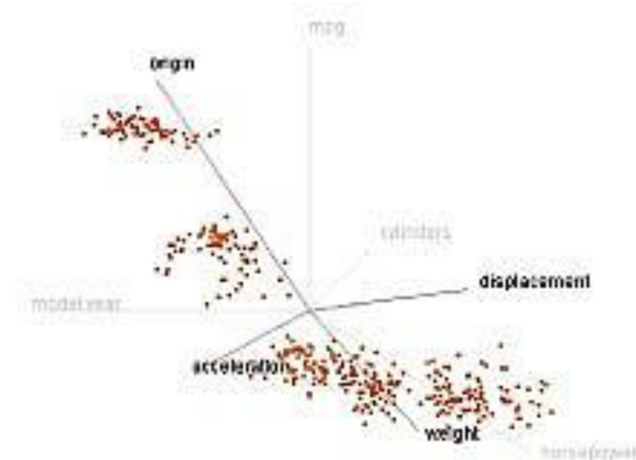
- no prior PCA and subsequent projection
- instead, a point P is plotted as a vector sum of all axis coordinates

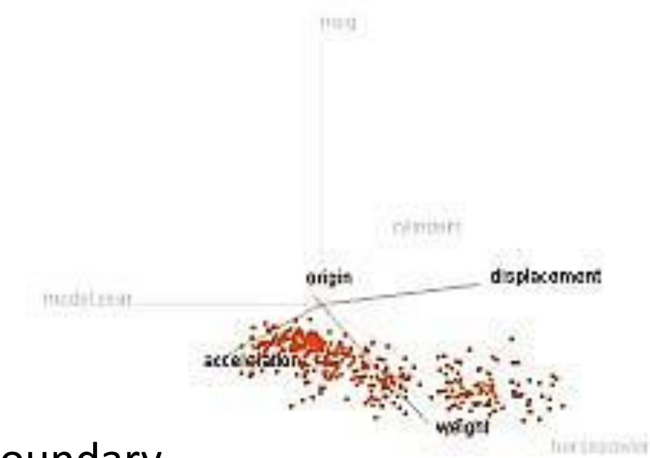


Star Coordinates

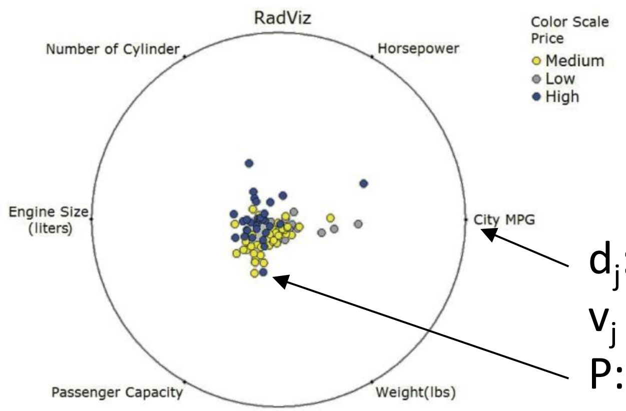
Operations defined on Star Coords

- scaling changes contribution to resulting visualization
- axis rotation can visualize correlations
- also used to reduce projection ambiguities





Similar paradigm: RadViz



$$P = \sum_{i=1}^{n} w_j v_j$$

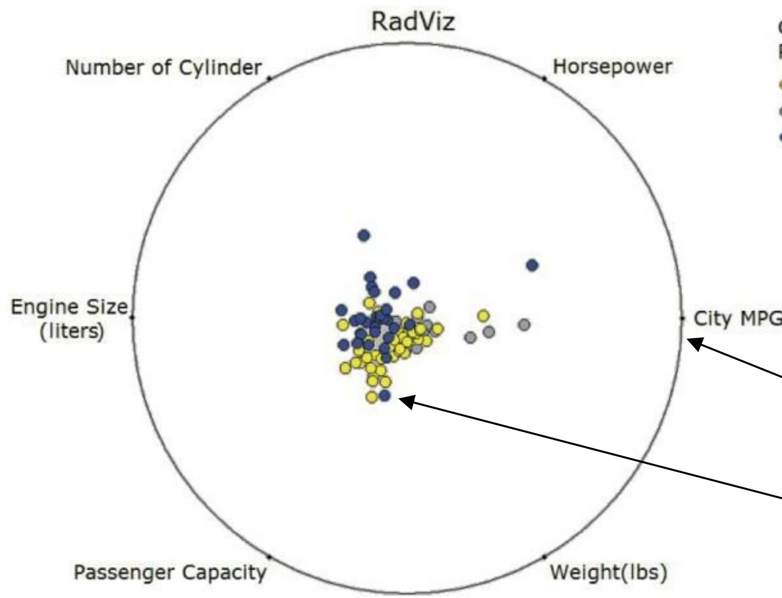
$$w_i = d_j / \sum_{k=1}^{n} d_k$$

d_j: attribute coordinates on disk boundary

v_j : data vector values

P: point location in RadViz disk

RadViz



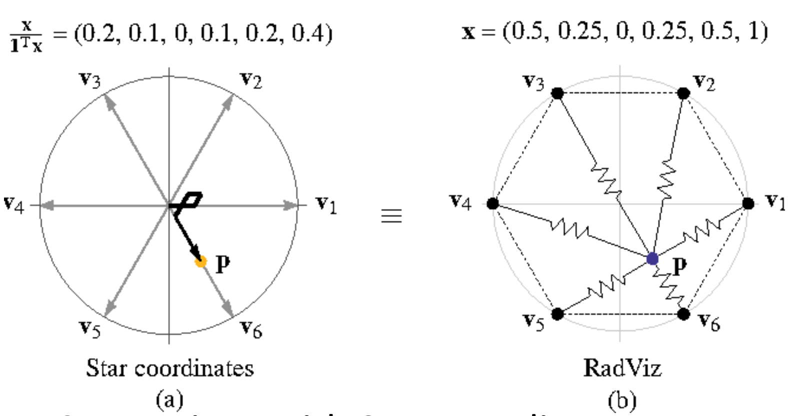
$$P = \sum_{i=1}^{n} w_j v_j$$

$$w_i = d_j / \sum_{k=1}^{n} d_k$$

v_j: attribute coordinates on disk boundary

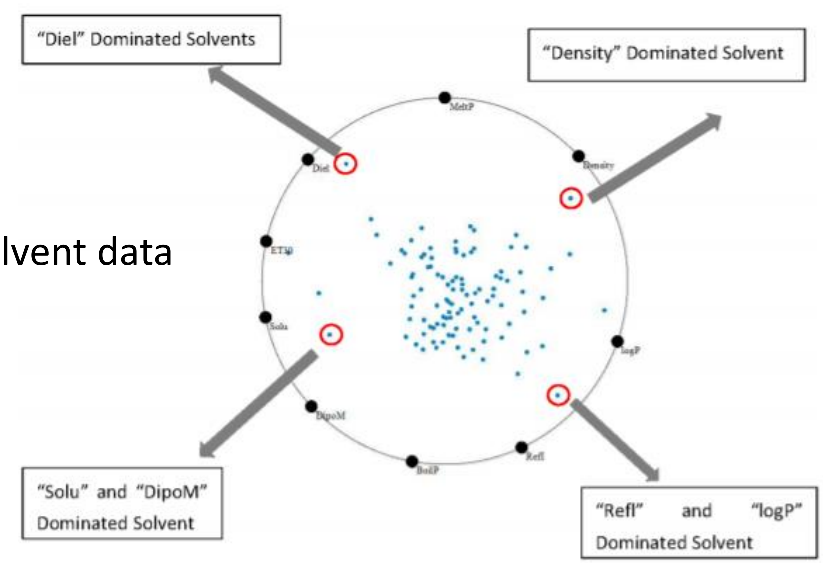
d_j : data vector values

P: point location in RadViz disk



Comparison with Star-coordinates

Solvent data



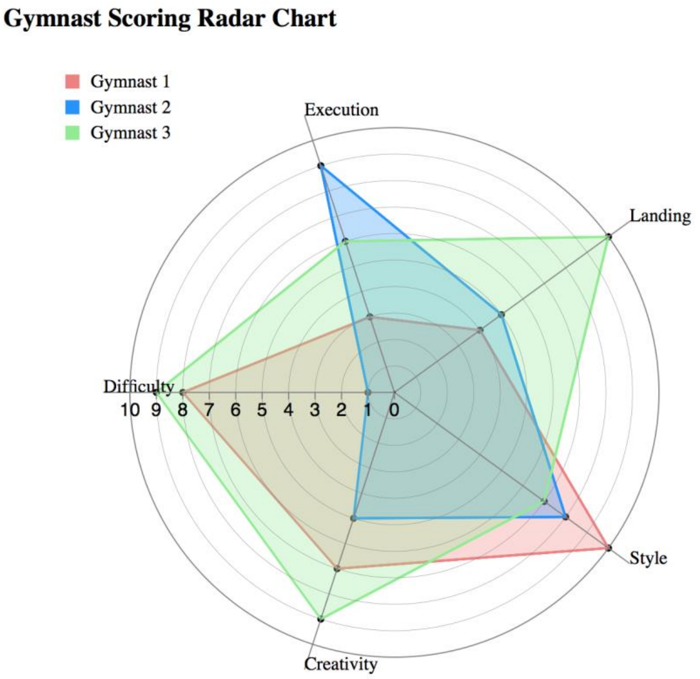
Radar Chart

Equivalent to a parallel coordinates plot, with the axes arranged radially

- each star represents a single observation
- can show outliers an commonalities nicely

Disadvantages

- hard to make trade-off decisions
- distorts data to some extents when lines are filled in



Commonalities

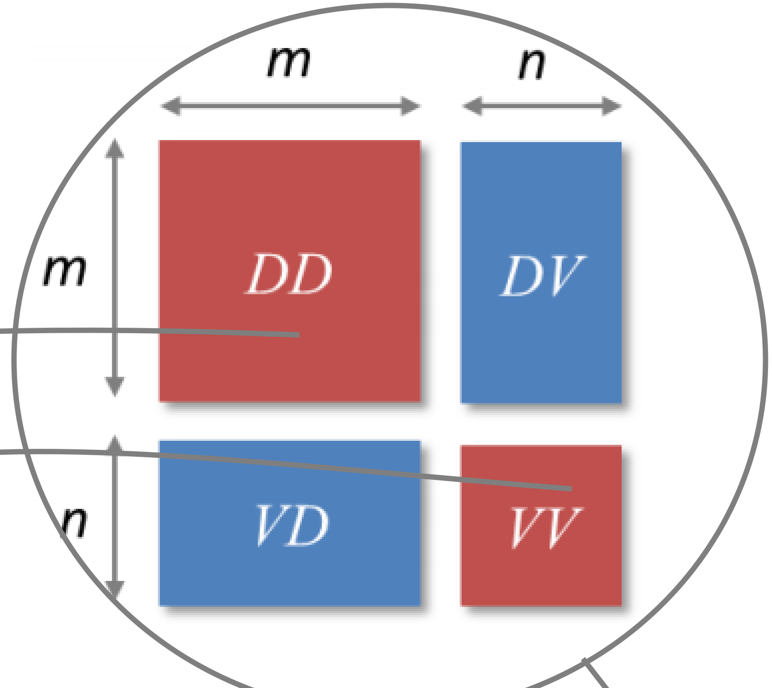
All of these radial scatterplot displays share the following characteristics

- allow users to see the data points in the context of the variables
- but can suffer from projection ambiguity
- some offer interaction to resolve some of these shortcomings
- but interaction can be tedious

Are there visualization paradigms that can overcome these problems?

- yes, algorithms that optimize the layout to preserve distances or similarities in high-dimensional space
- what is this algorithm?
- yes, MDS (Multi-Dimensional Scaling)
- we have discussed MDS before (so we will skip further discussion)

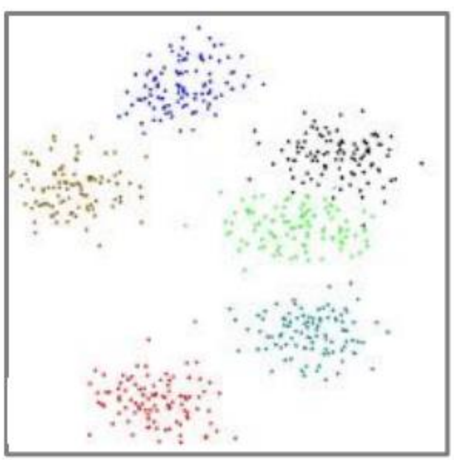
Uses of MDS



data similarity matrix (DD)

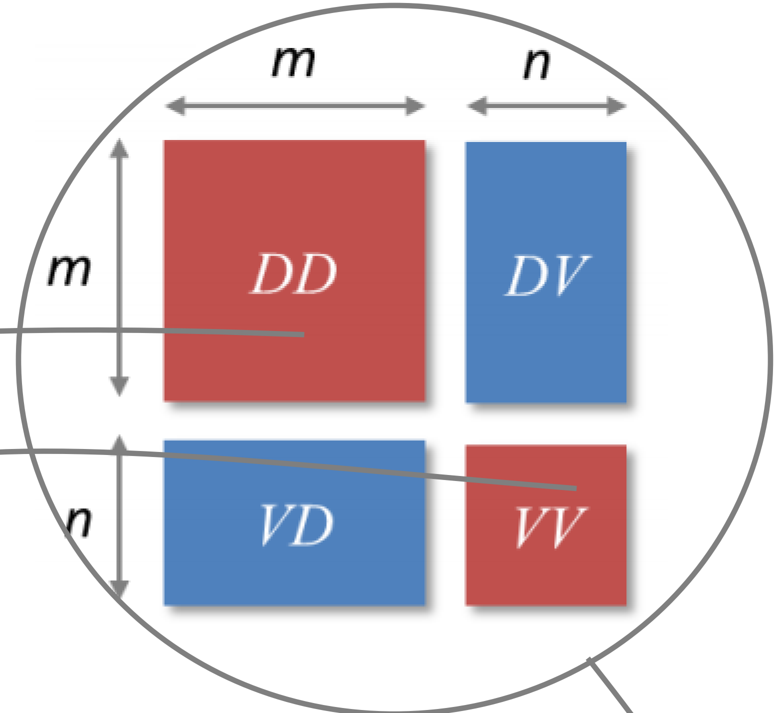
attribute similarity matrix (VV)

Data layout



+ data-attribute similarity matrix (VD, DV)

Uses of MDS



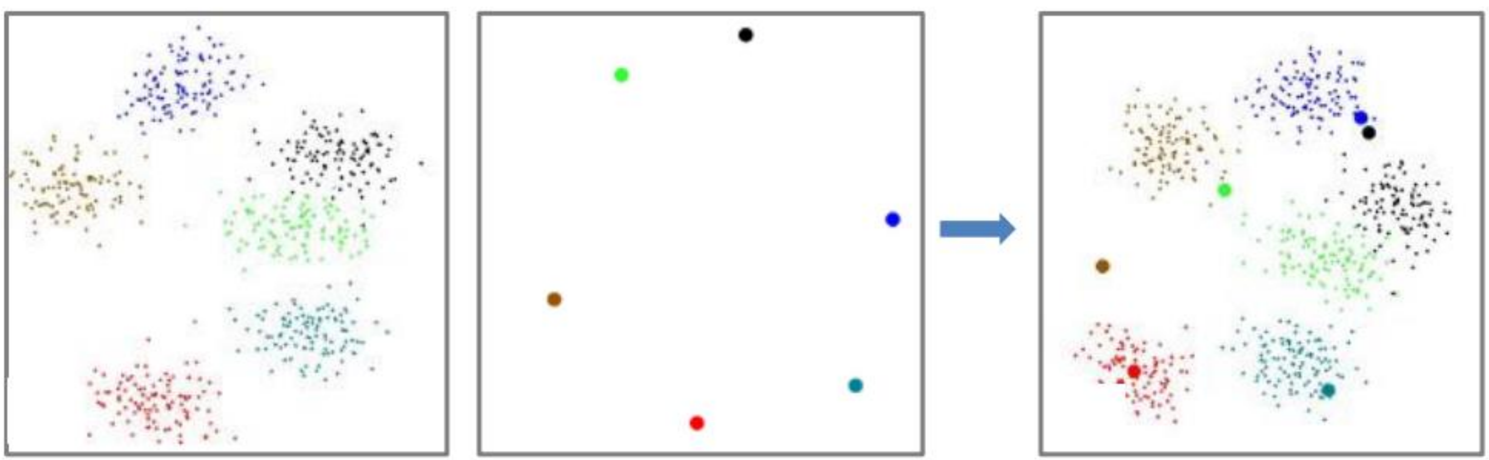
data similarity matrix (DD)

attribute similarity matrix (VV)

Data layout

Attribute layout

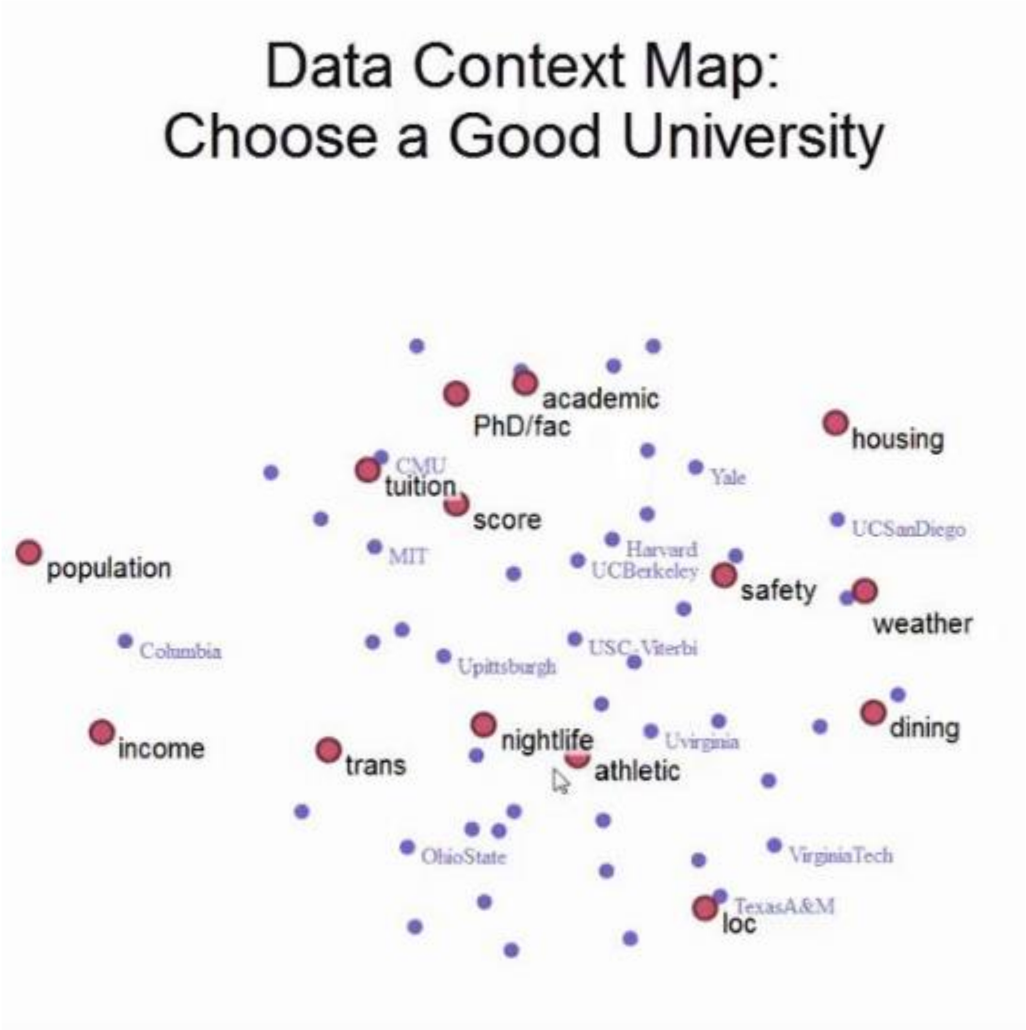
Combined layout



+ data-attribute similarity matrix (VD, DV)

Yields the Data Context Map

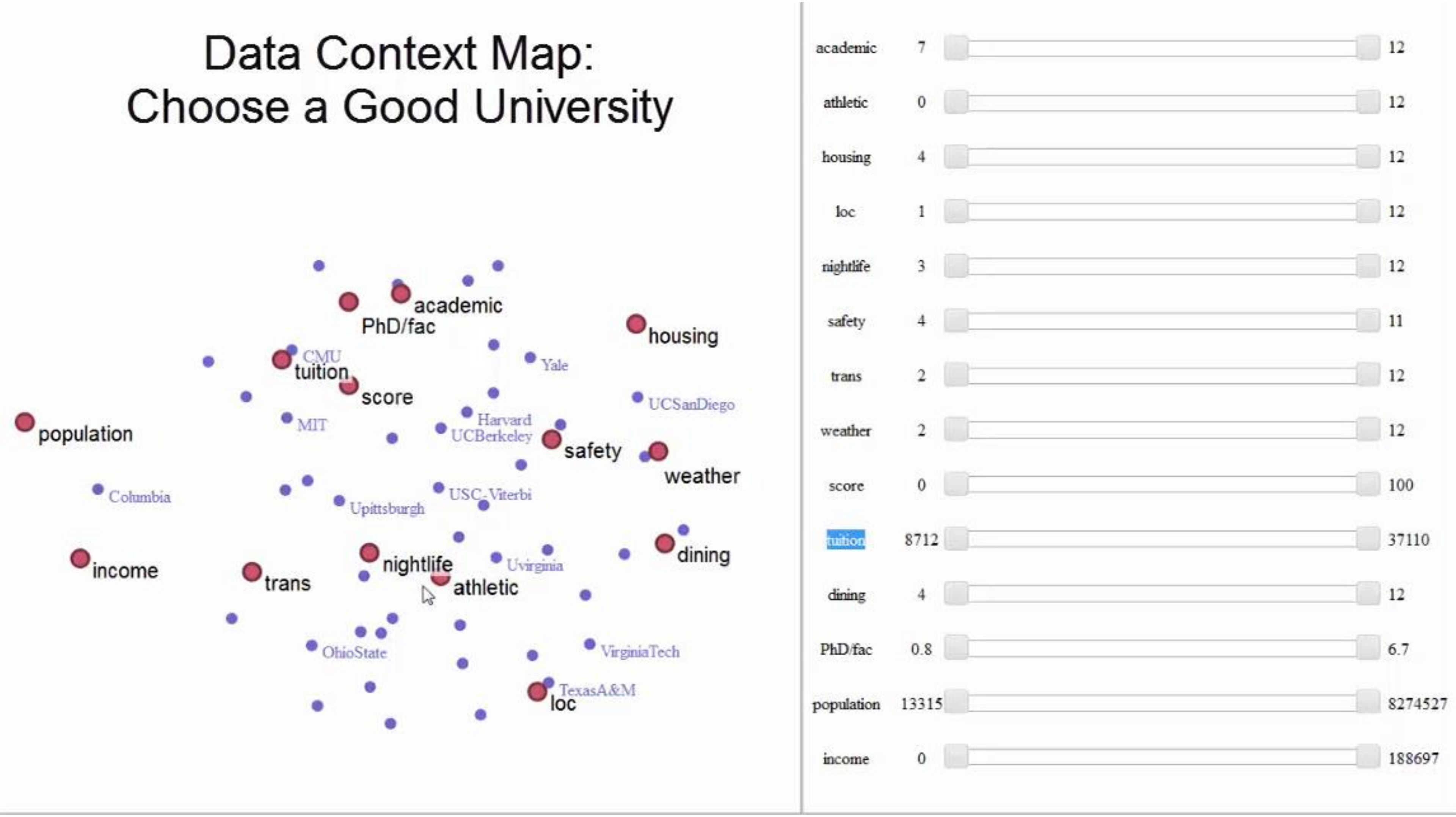
Data visualized in the context of the attributes



S. Cheng, K. Mueller, “The Data Context Map: Fusing Data and Attributes into a Unified Display,” *IEEE Trans. on Visualization and Computer Graphics*, 22(1): 121-130, 2016.

[youtube]

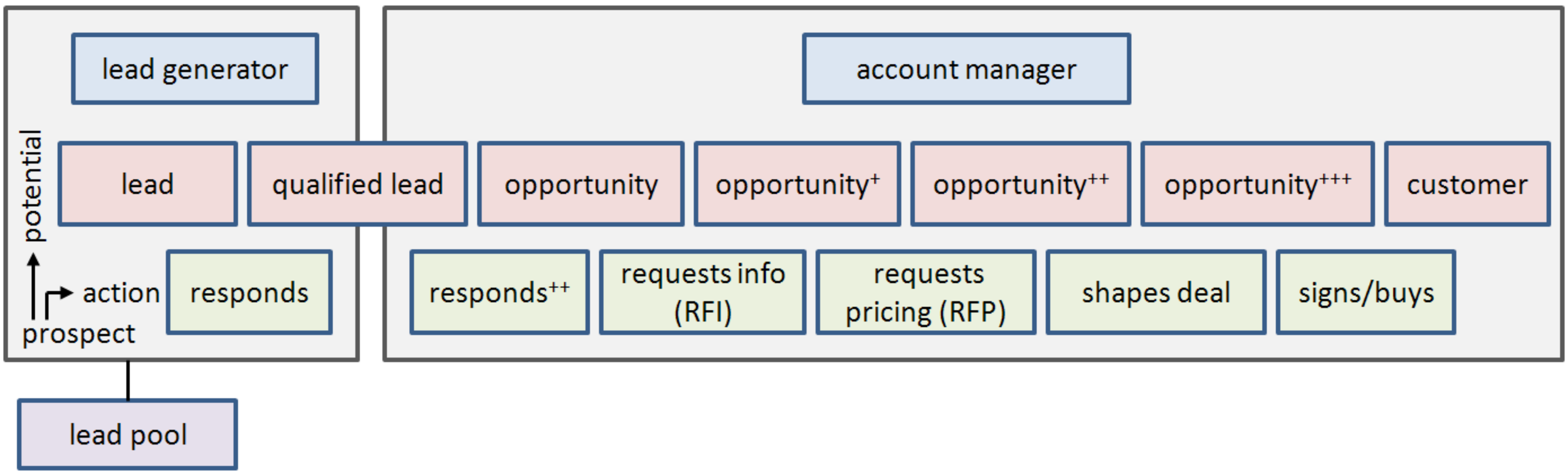
Data Context Map in Action



Telling Stories with Parallel Coordinates

Example: Sales Strategy Analysis

Anatomy of a Sales Pipeline



The Setup

Scene:

- a meeting of sales executives of a large corporation, Vandelay Industries

Mission:

- review the strategies of their various sales teams

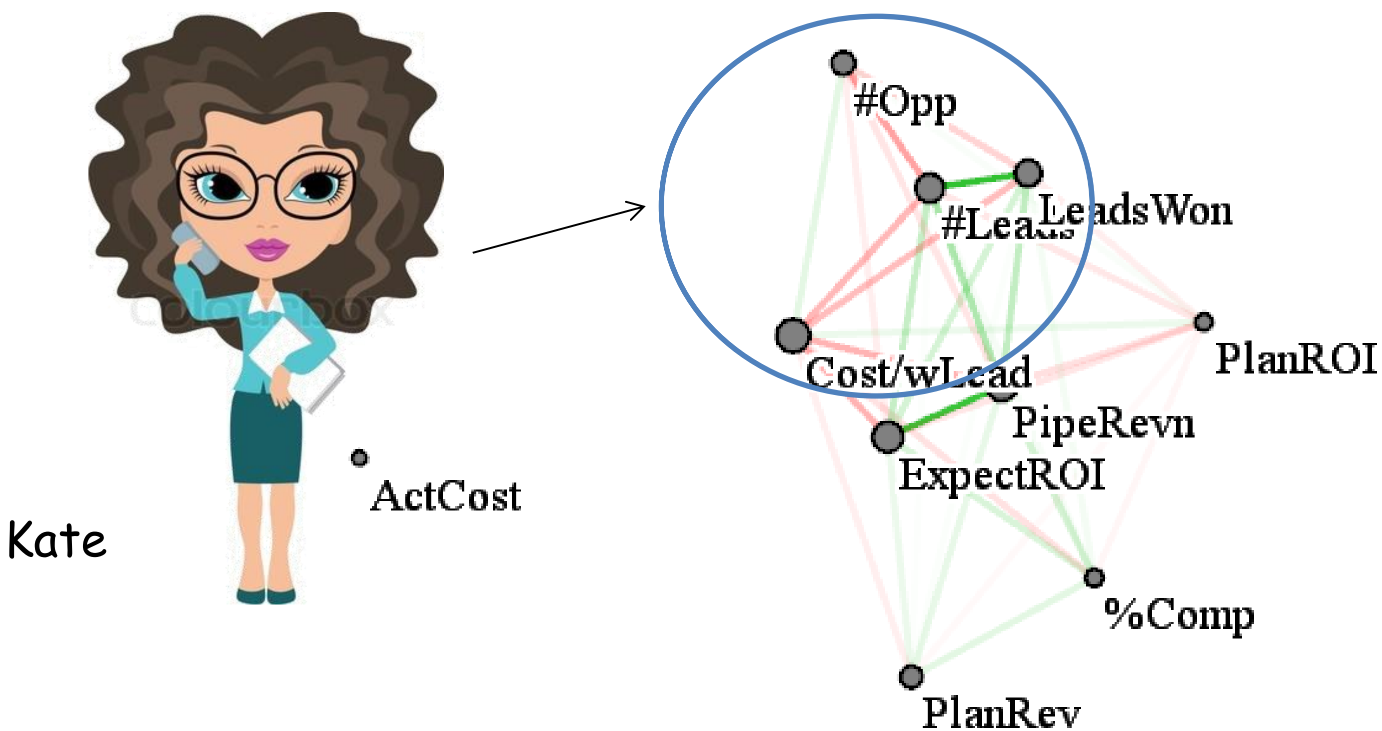
Evidence:

- data of three sales teams with a couple of hundred sales people in each team

Kate Explains It All

Meet Kate, a sales analyst in the meeting room:

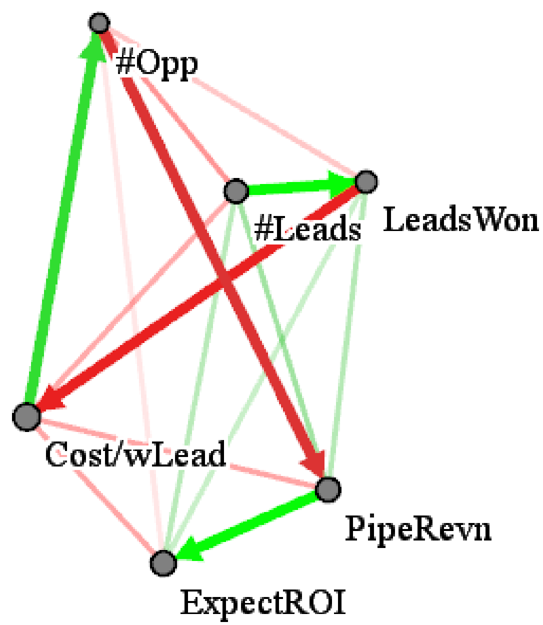
> "OK…let's see, cost/won lead is nearby and it has a positive correlation with #opportunities but also a negative correlation with #won leads"

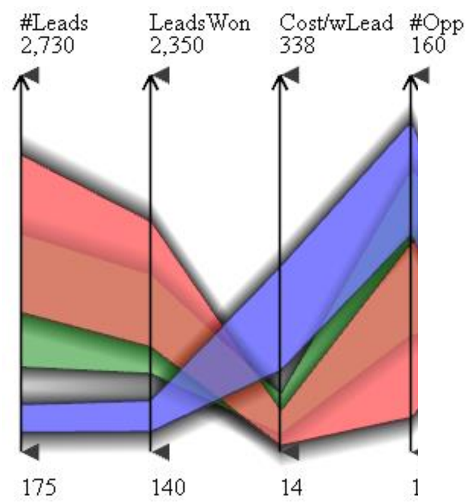


Kate Designs the Narration

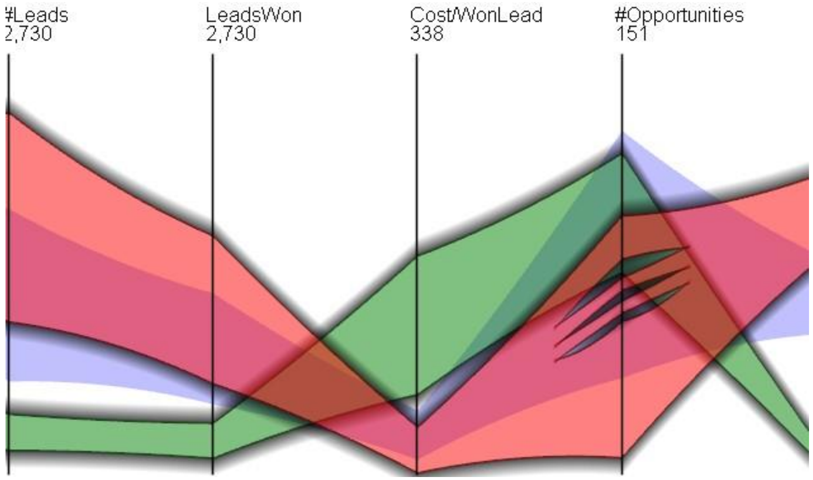
"Let's go and make a revealing route!"

- she uses the mouse and designs the route shown
- she starts explaining the data like a story …





Further Insight



Kate notices something else:

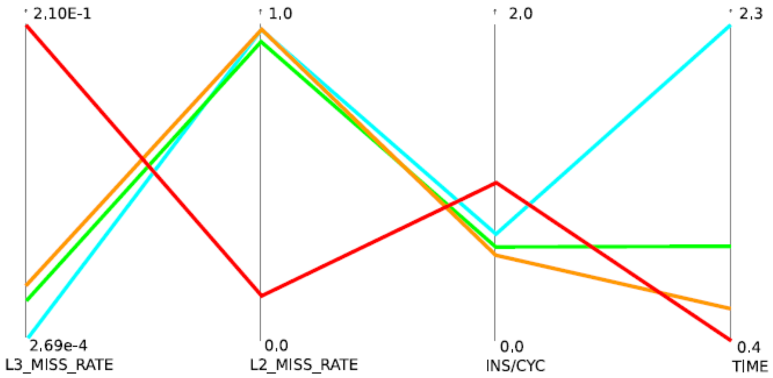
- now looking at the red team
- there seems to be a spread in effectiveness among the team
- the team splits into three distinct groups

She recommends: "Maybe fire the least effective group or at least retrain them"

Recent Reviewer Comment

From a paper sent to a software visualization conference:

Figure 8

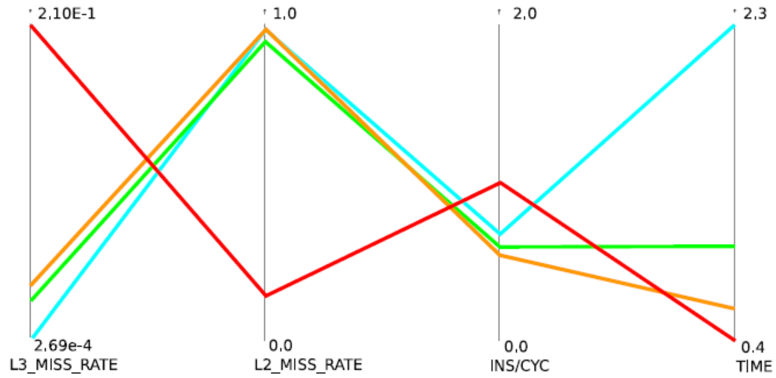


- Multiple visualizations appear to present categorical data as line graphs, which seems a strange choice.

Recent Reviewer Comment

From a paper sent to a software visualization conference:

Figure 8

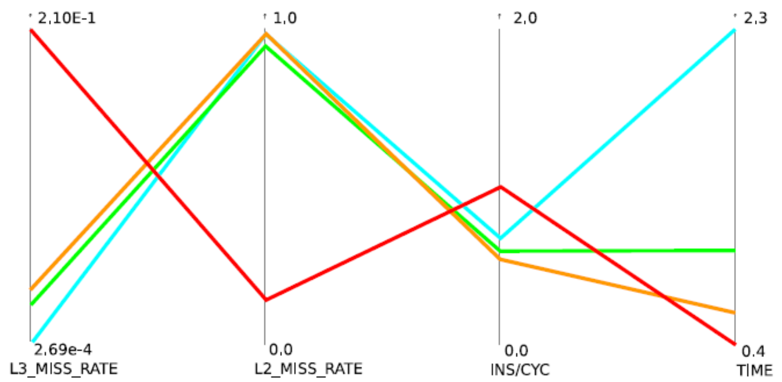


- Multiple visualizations appear to present categorical data as line graphs, which seems a strange choice. Figure 8, for example, at first sight appeared to be showing a change over time, but in fact further inspection shows that the different x-coordinates are almost entirely unrelated to one another and in no particular order.

Recent Reviewer Comment

From a paper sent to a software visualization conference:

Figure 8



- Multiple visualizations appear to present categorical data as line graphs, which seems a strange choice. Figure 8, for example, at first sight appeared to be showing a change over time, but in fact further inspection shows that the different x-coordinates are almost entirely unrelated to one another and in no particular order. This is such an unusual choice that I'm not sure that I am understanding the role of the graphs correctly.

How to Teach Mainstream Users

Learning Visualizations by Analogy

Puripant Ruchikachorn and Klaus Mueller

Stony Brook University

https://www.youtube.com/watch?v=mdolkHA-RpA

User Studies

Encode user responses based on task complexities

▪ none (0):	cannot report any findings
▪ low (1):	understand representation visual encoding
▪ medium (2):	identify groups and outliers
▪ high (3):	recognize correlations and trends

User Studies – Car Dataset

Visual understanding:

(1) The MPG of the orange-highlighted car is ~40% of its range

(2) There is just one line at the top of the acceleration scale

(3) Heavier cars are faster

Data Understanding:

(1) The number of cylinders of the orange-highlighted car is 4, one fifth between 3 and 8.

(2) Many cars have the same numbers of cylinders, mostly even numbers particularly 4 and 8.

(3) Heavier cars have more cylinders and hence more horsepower and speed.

Results

Participants		V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11
Parallel Coordinates Plot	Before	3	0	0	0	1	0	2	1	0	3	3
	After	3	2	2	1	2	2	3	2	1	3	3
	Diff.	0	2	2	1	1	2	1	1	1	0	0

D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11
0	2	3	1	1	3	1	1	2	0	3
2	3	3	3	1	3	2	2	3	2	3
2	1	0	2	0	0	1	1	1	2	0

Plot Selection

Scatterplot Matrix

Scatterplot version of parallel coordinates

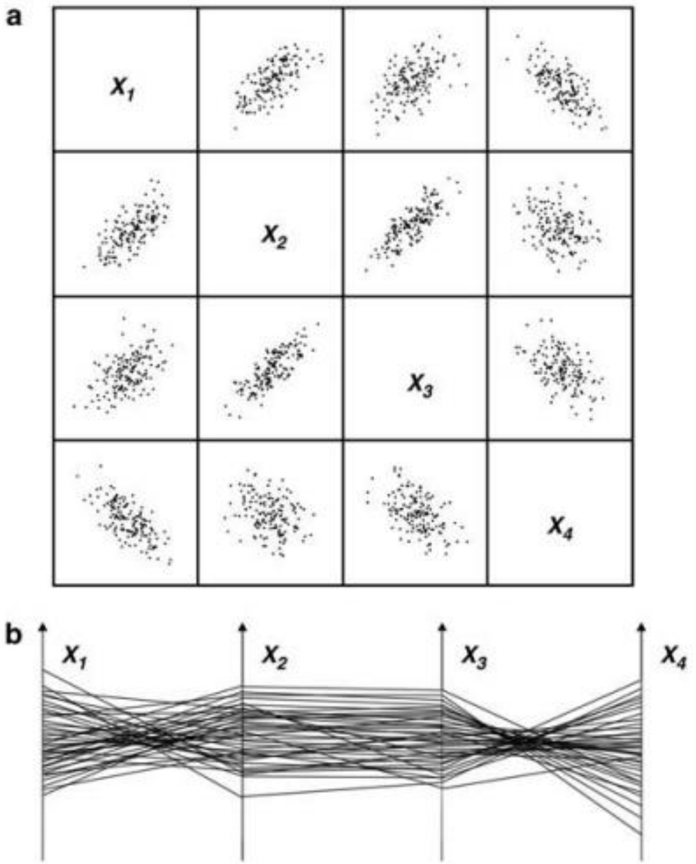
- distributes n(n-1) bivariate relationships over a set of tiles
- for n=4 get 16 tiles
- can use n(n-1)/2 tiles

For even moderately large n:

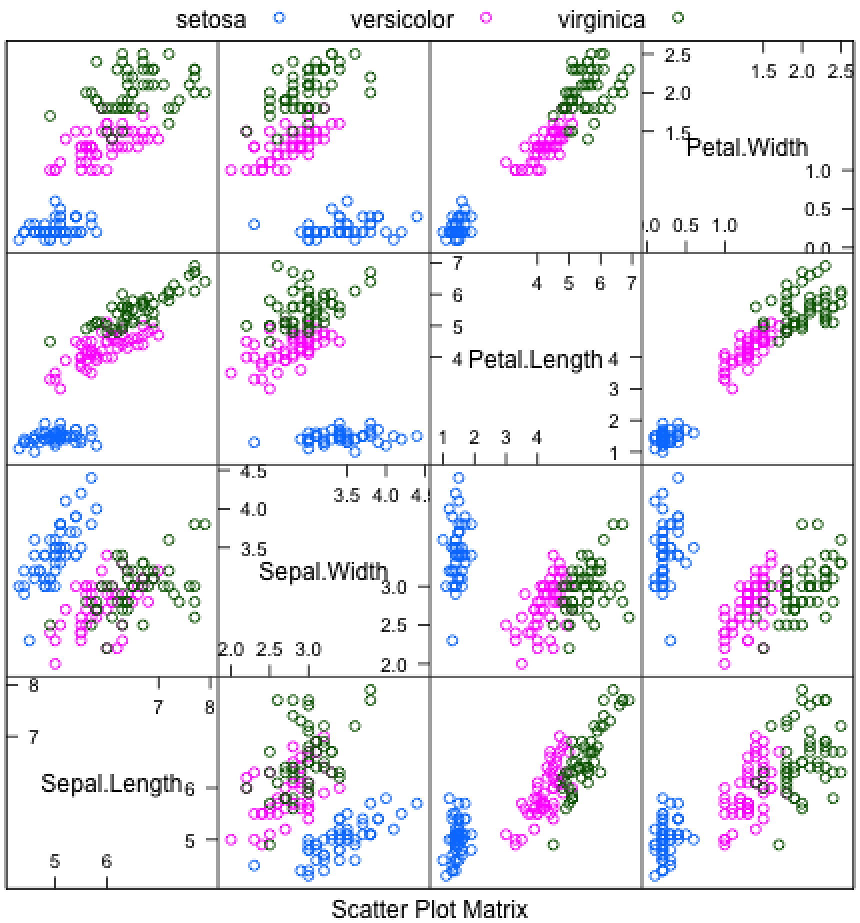
- there will be too many tiles

Which plots to select?

- plots that show correlations well
- plots that separate clusters well



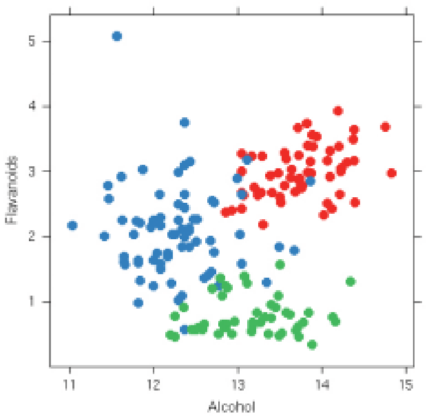
Scatterplot Matrix



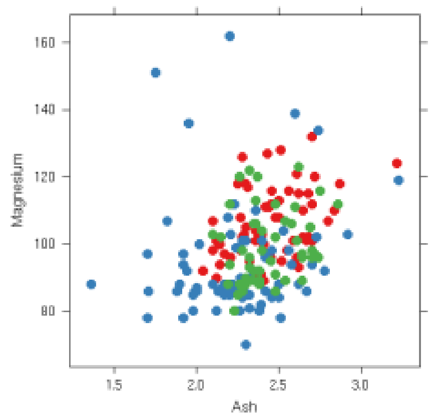
Select the most interesting tiles and show them to the user

AUTOMATED SCATTERPLOT SELECTION

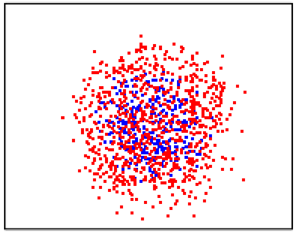
Several metrics, a good one is Distance Consistency (DSC)

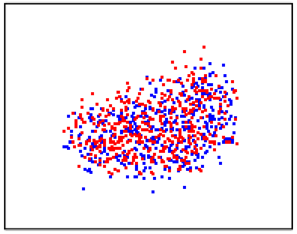


(a) **DSC=90**

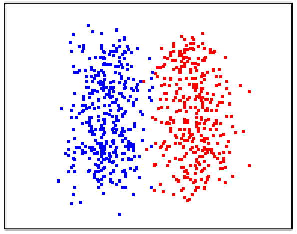


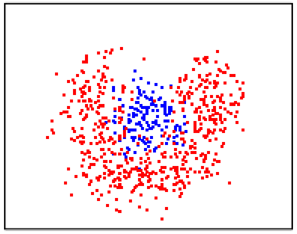
(b) **DSC=49**


(d) 29


(e) 15

bad


(a) 99


(b) 74

OK

$$\mathbf{DSC} = \frac{\left|x' \in v(X) : \mathbf{CD}(x', centr'(c_{clabel(x)}) = true\right|}{k}$$

- measures how "pure" a cluster is
- pick the views with highest normalized DSC

M. Sips et al., Computer Graphics Forum, 28(3): 831–838, 2009

Dunn Index

Favors clusters that (1) are compact and (2) are well isolated

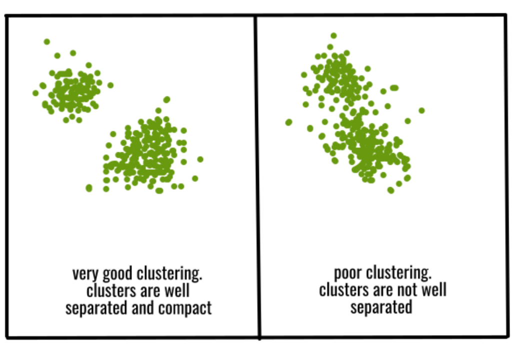
$$DI_m = \frac{\min\limits_{1 \leqslant i < j \leqslant m} \delta(C_i, C_j)}{\max\limits_{1 \leqslant k \leqslant m} \Delta_k}.$$

min separation

max spread

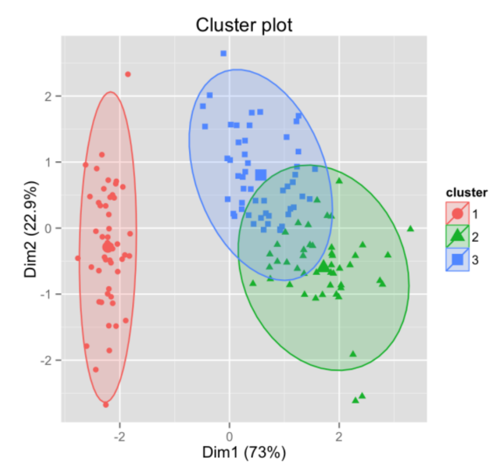
$\Delta_i = \frac{\sum_{x \in C_i} d(x, \mu)}{|C_i|}, \mu = \frac{\sum_{x \in C_i} x}{|C_i|}$, calculates distance of all the points from the mean.

$\delta(C_i, C_j)$ be this intercluster distance metric, between clusters C_i and C_j.

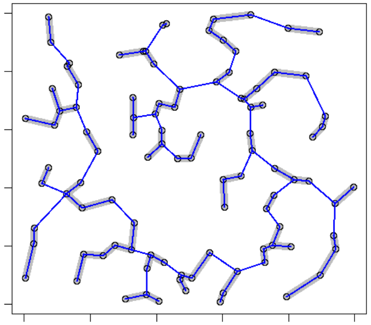


high Dunn Index low Dunn Index

determine the quality of k-means clustering

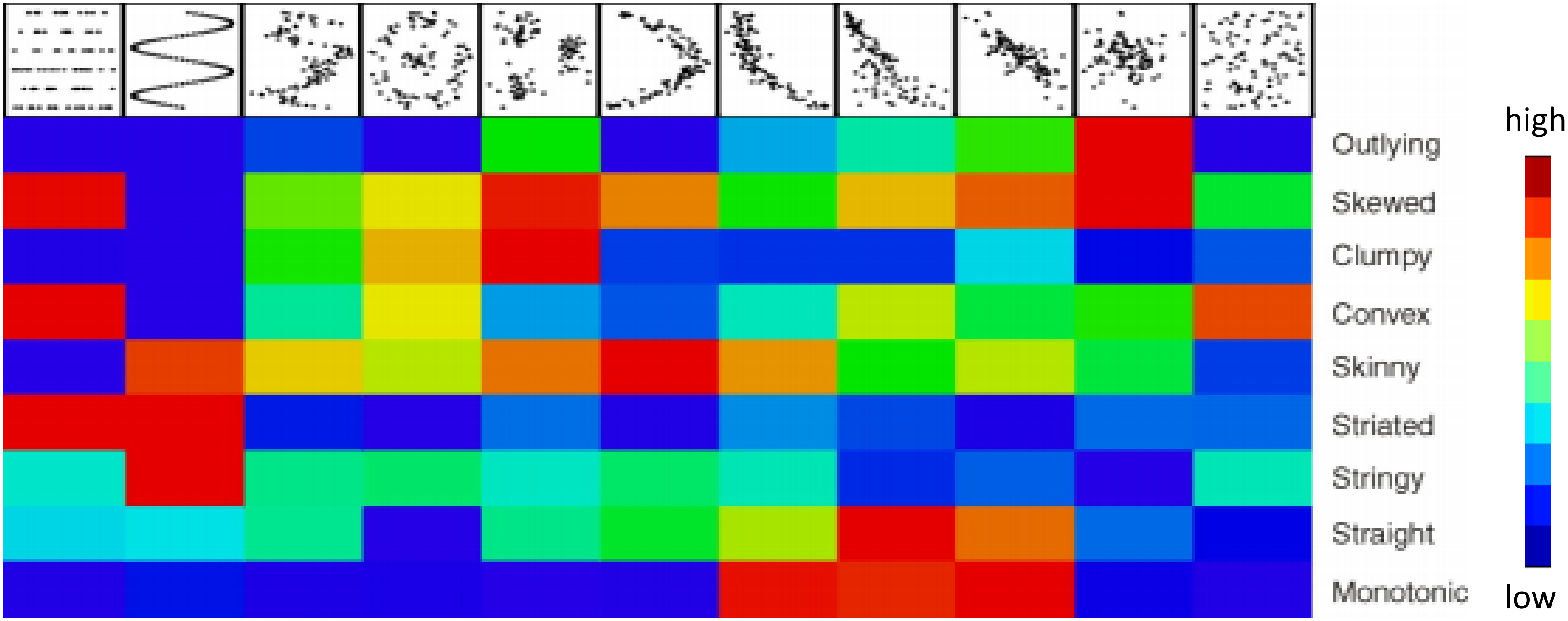


Scagnostics



Describe scatterplot features by graph theoretic measures

- mostly built on minimum spanning tree
- can be used to summarize large sets of scatterplots



Scatterplot of Scatterplots

Use scagnostics to quickly survey 1,000s of scatterplots

- compute scagnostics measures
- create scatterplot matrix of these measures
- each scatterplot is a point