CSE 332
Introduction to Visualization

High-Dimensional Data

Klaus Mueller

Computer Science Department
Stony Brook University

Lecture	Topic	Projects
1	Intro, schedule, and logistics	
2	Applications of visual analytics, data types	
3	Data sources and preparation	Project 1 out
4	Data reduction, similarity & distance, data augmentation	
5	Dimension reduction	
6	Introduction to D3	
7	Visual communication using infographics	
8	Visual perception and cognition	Project 2 out
9	Visual design and aesthetic	
10	D3 hands-on presentation	
11	Cluster analysis	
12	Visual analytics tasks and design	
13	High-dimensional data VIS: linear projections	Project 3 out
14	High-dimensional data VIS: optimized layouts	
15	Visualization of spatial data	
16	Midterm	
17	Illumination and isosurface rendering	
18	Scientific visualization	
19	Non-photorealistic and illustrative rendering	Project 4 out
20	Midterm discussion	
21	Principles of interaction	
22	Visual analytics and the visual sense making process	
23	Visualization of graphs and hierarchies	
24	Visualization of time-varying and streaming data	Project 5 out
25	Maps	
26	Memorable visualizations, visual embellishments	
27	Evaluation and user studies	
28	Narrative visualization, storytelling, data journalism, XAI	

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)

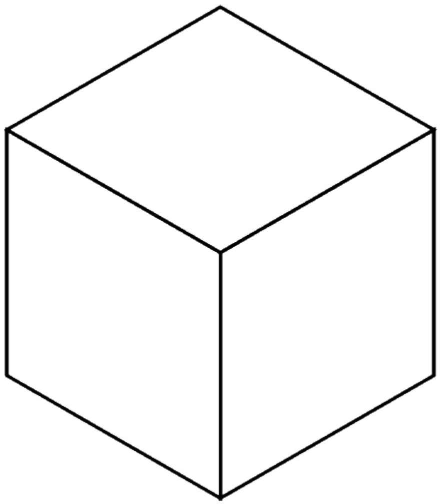
- the ability to create mental pictures or remember objects and people you have previously seen
- thought to be a vital precursor to creativity and abstract thinking

High-D Space is Tricky

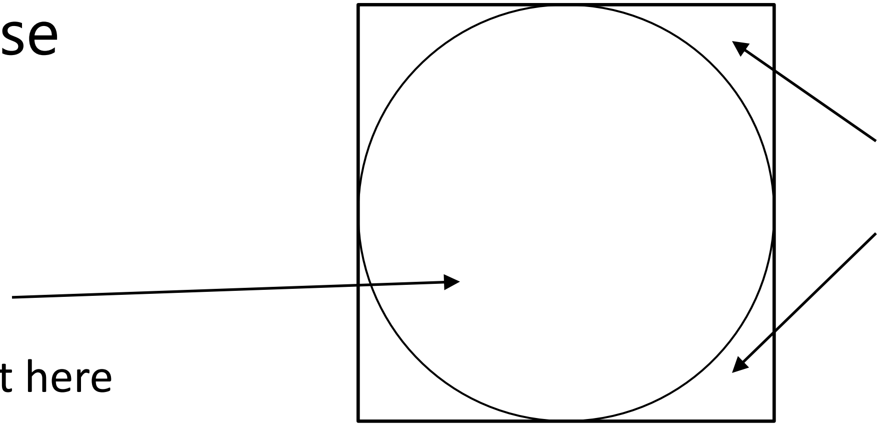
The curse of dimensionality

As $n \to \infty$

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

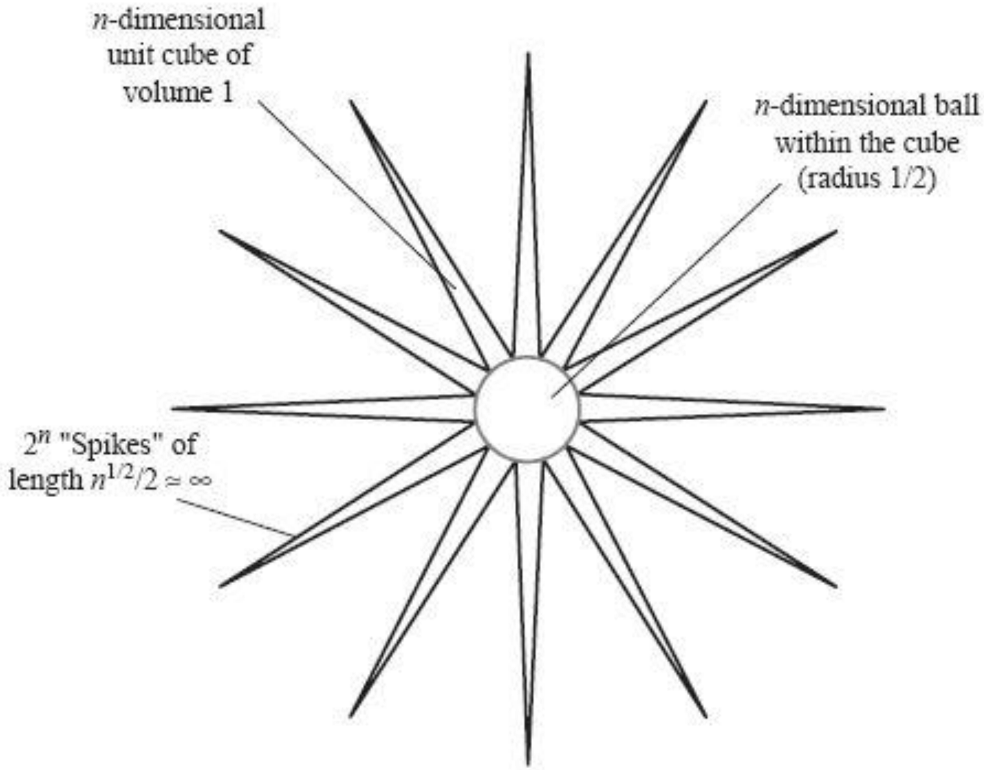


and very sparse



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

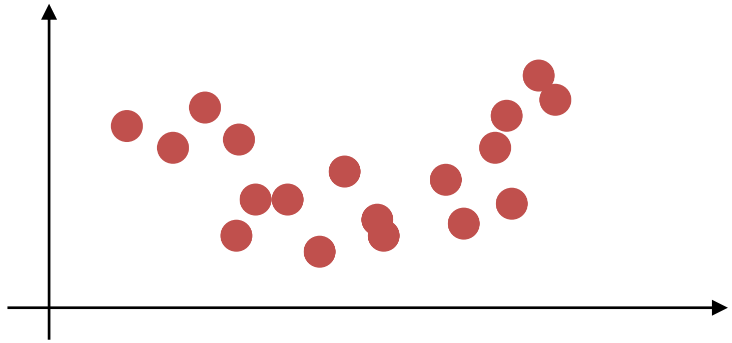
- with every extra dimension points get pulled apart further
- distances become meaningless

Sparseness Demonstration

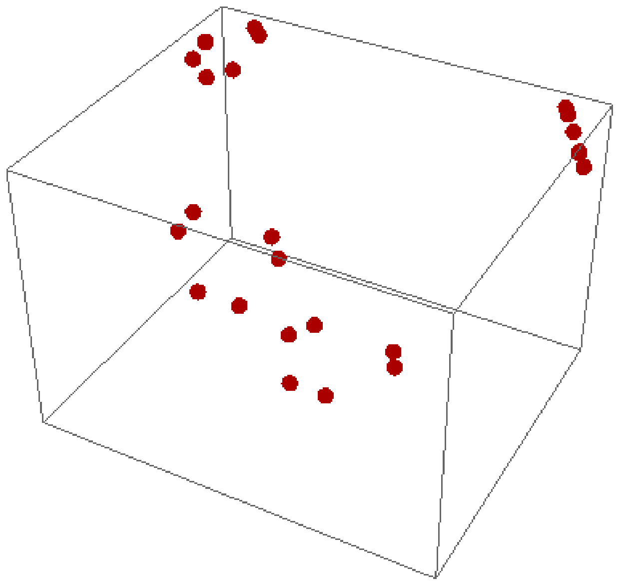
Space gets extremely sparse

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

1D – points are very close



2D – points spread apart



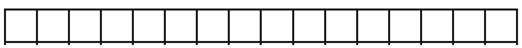
3D – getting even sparser

4D, 5D, … – sparseness grows further

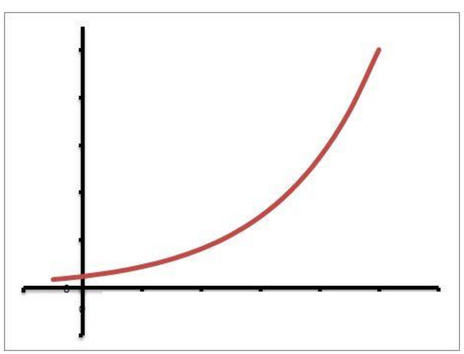
Space and Memory Management

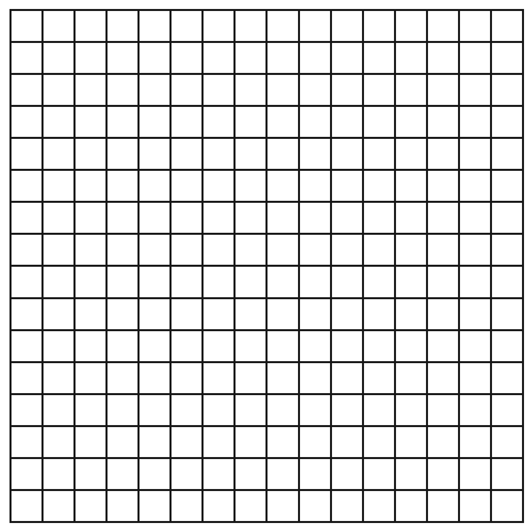
Indexing (and storage) also gets very expensive

- exponential growth in the number of dimensions

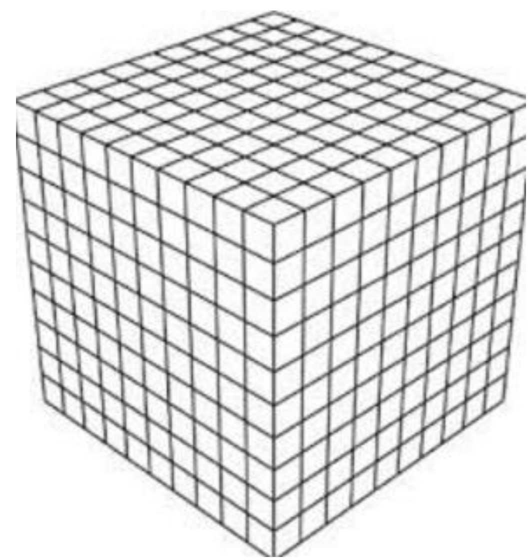


16 cells





16^2 = 56 cells

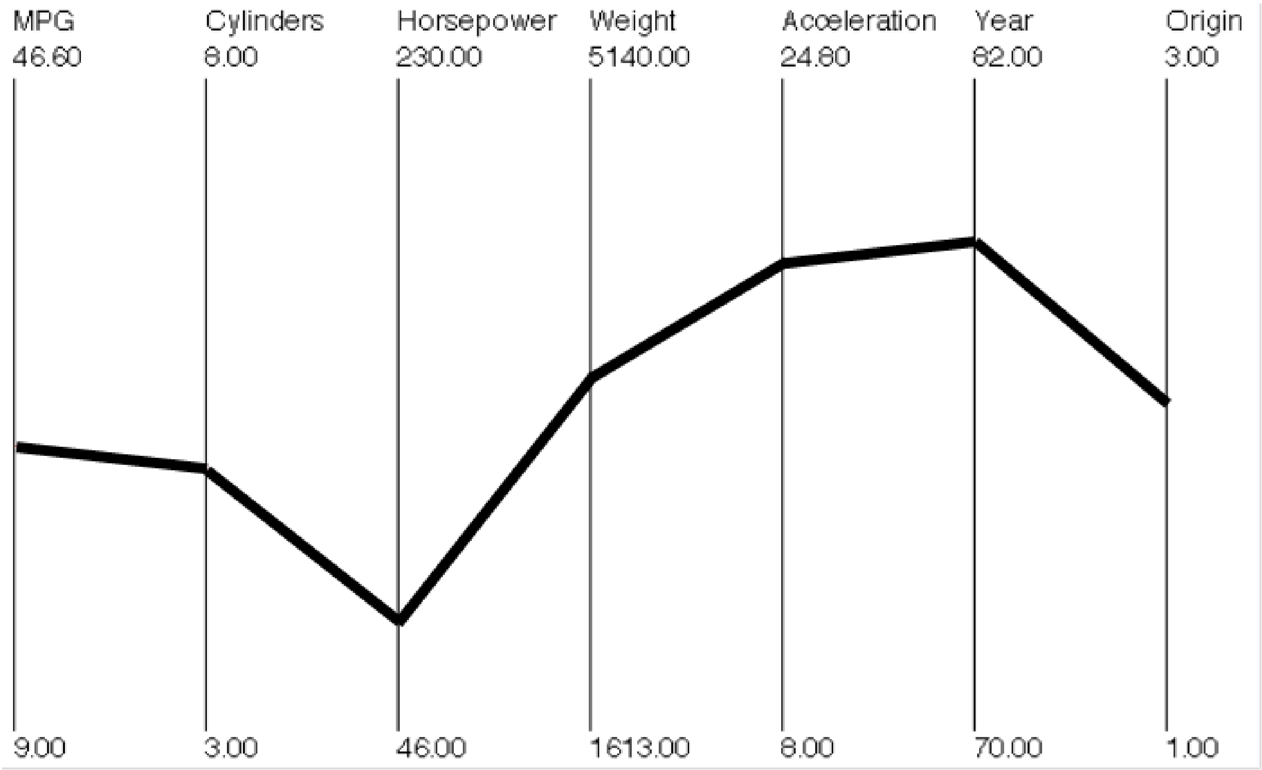


16^3 = 4,096 cells

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

Parallel Coordinates

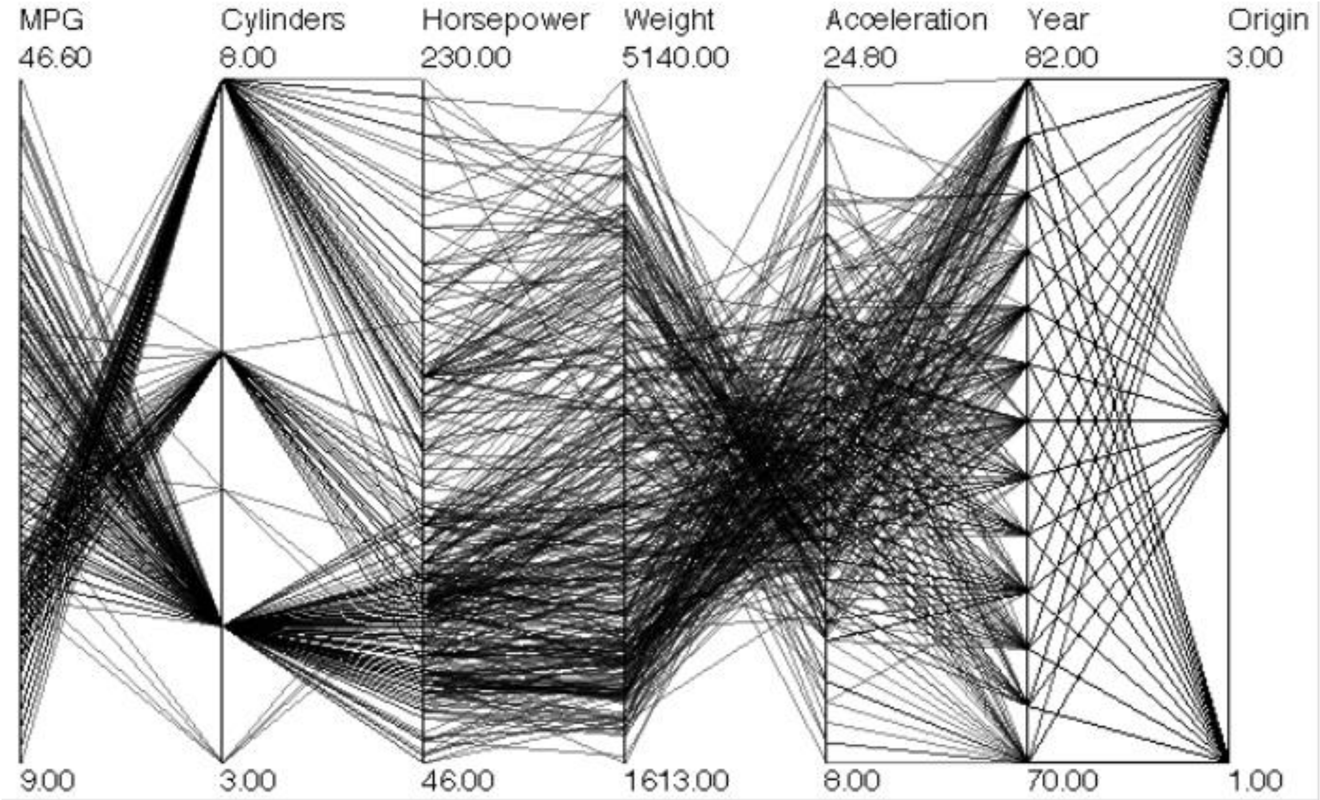
Parallel Coordinates – 1 Car



The N=7 data axes are arranged side by side

- in parallel

Parallel Coordinates – 100 Cars

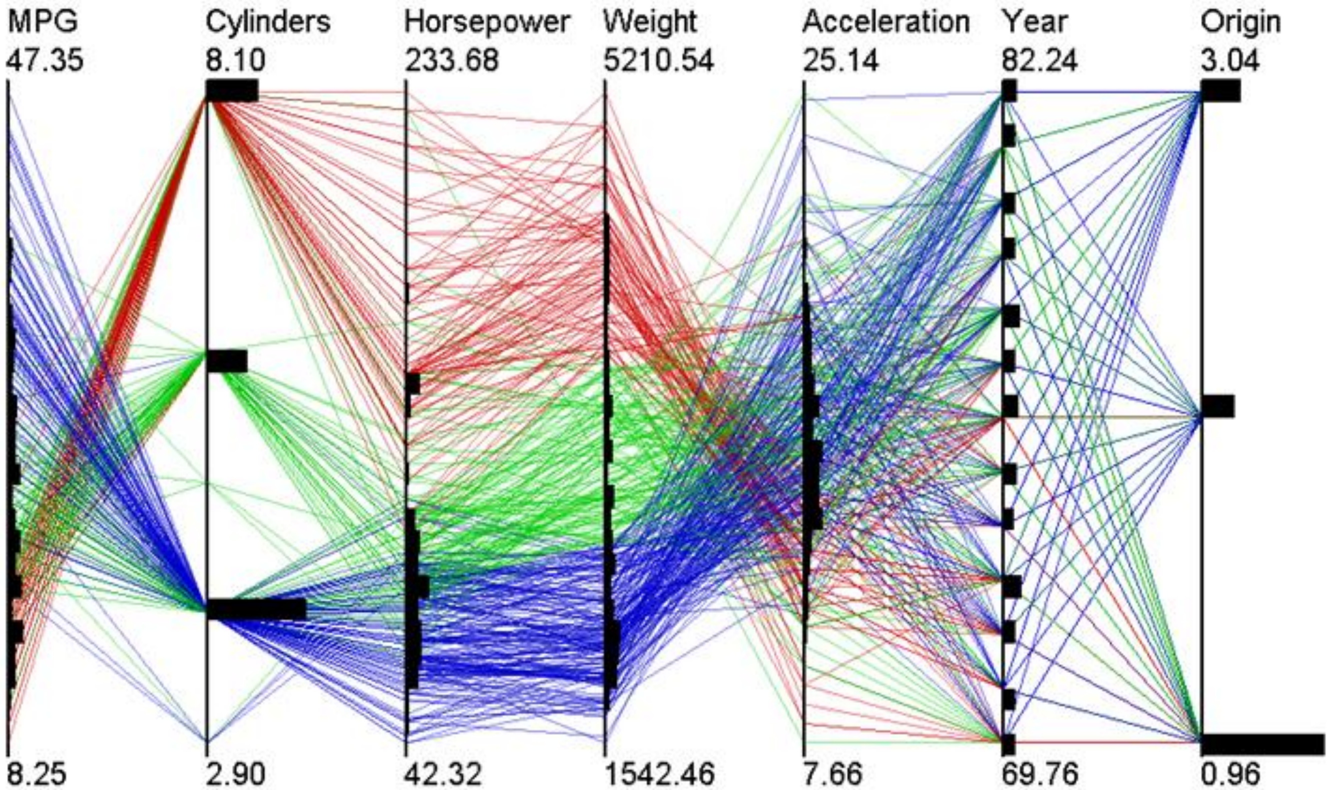


Hard to see the individual cars?

- what can we do?

socrative
by MasteryConnect

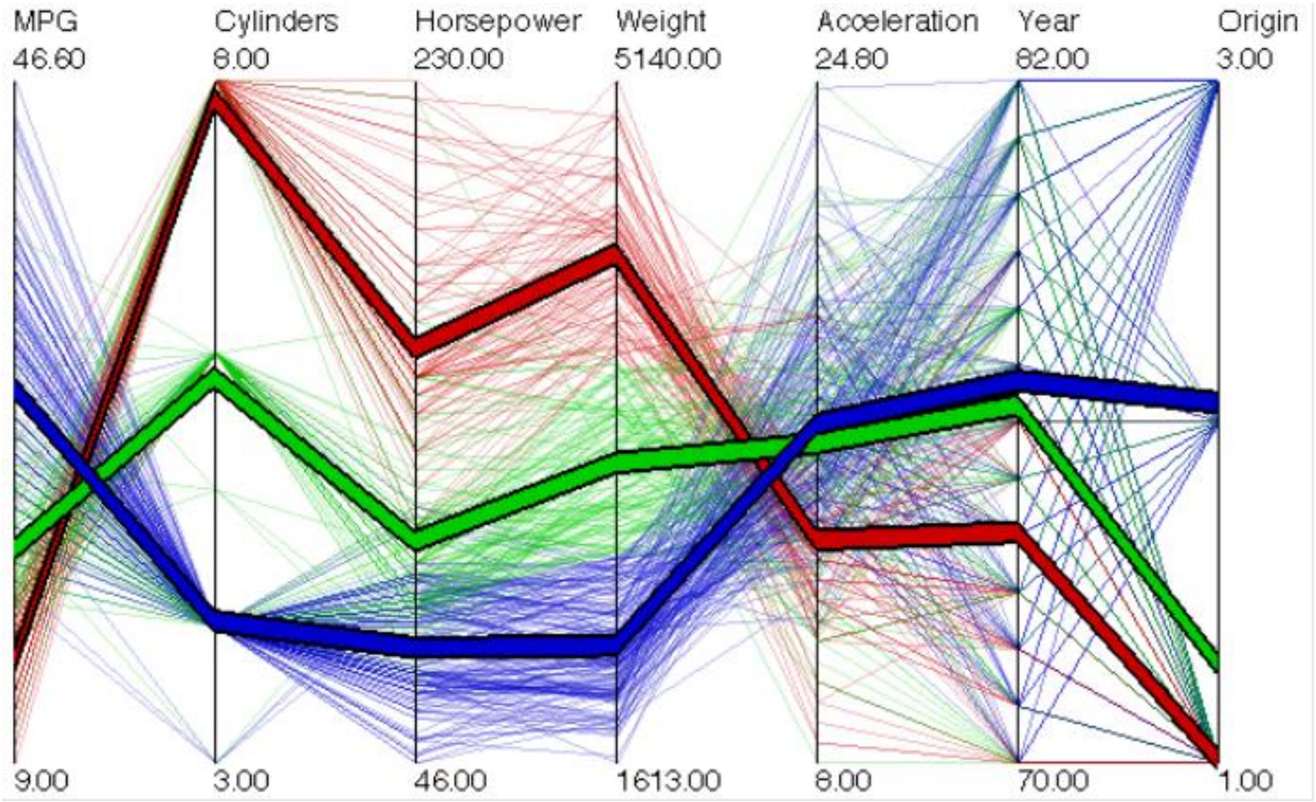
Parallel Coordinates – 100 Cars



Grouping the cars into sub-populations

- a clustering operation
- an be automated or interactive (put the user in charge)

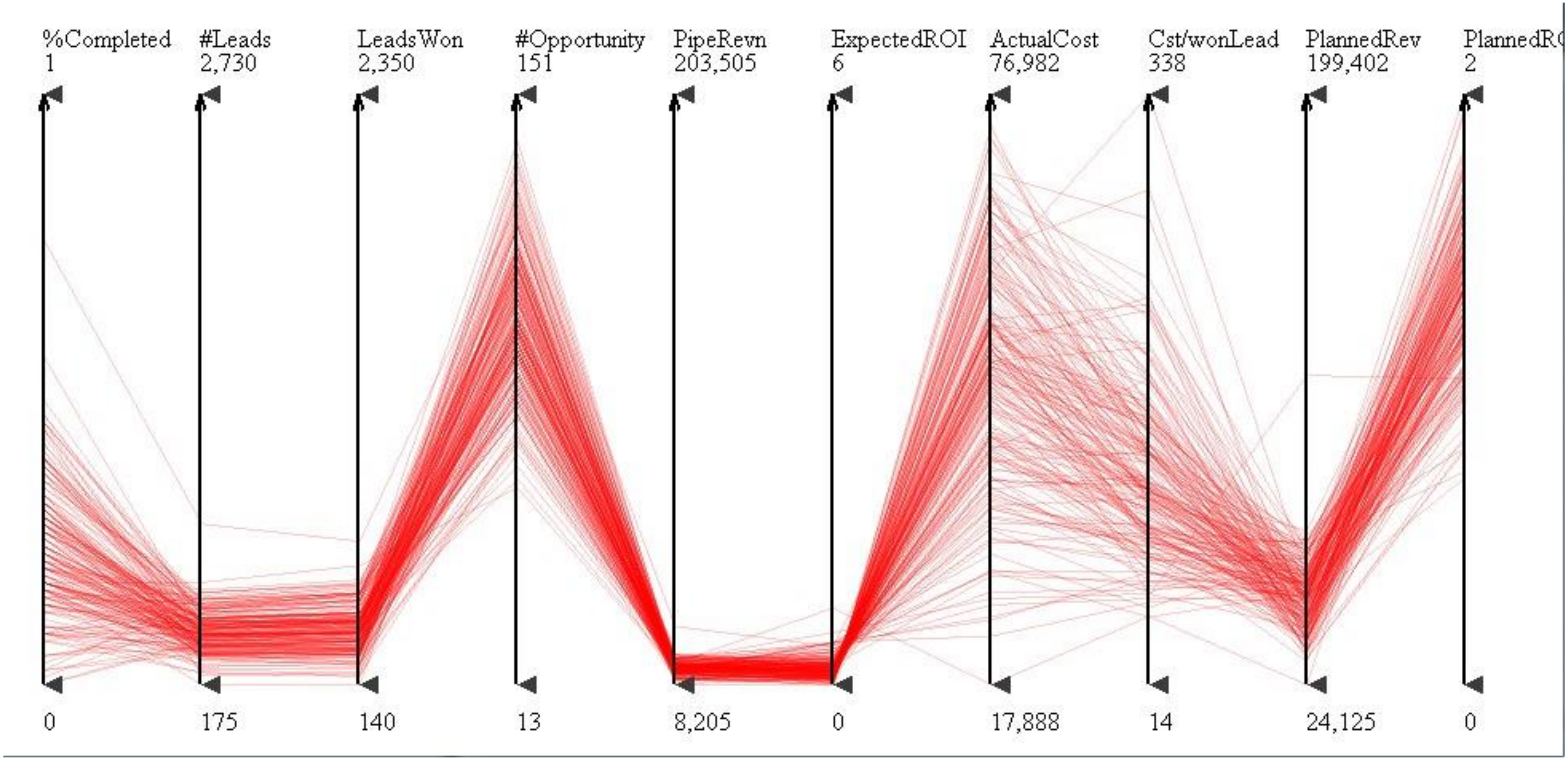
PC With Mean Trend



Computes the mean and superimposes it onto the lines

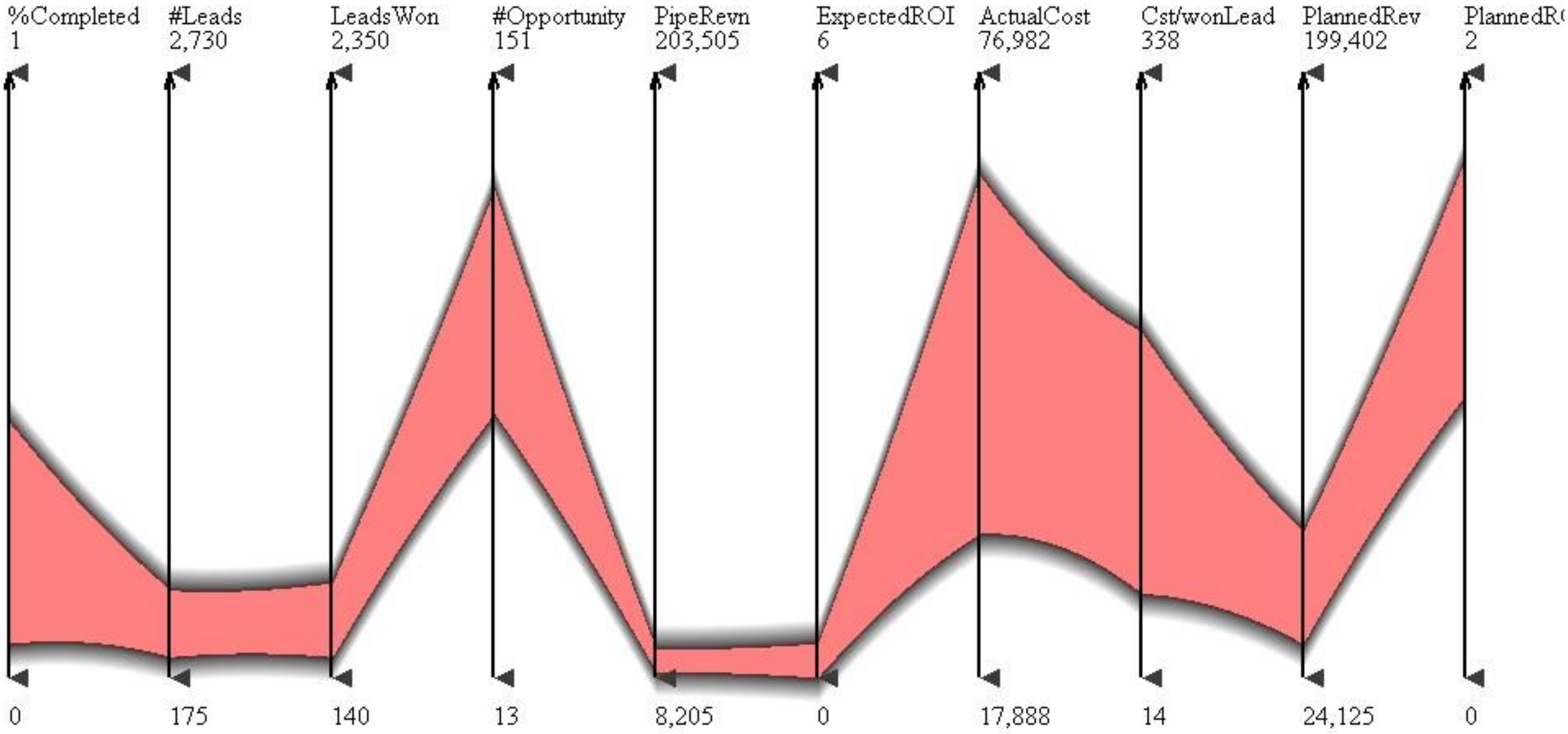
- allows one to see trends

PC With Illustrative Abstraction



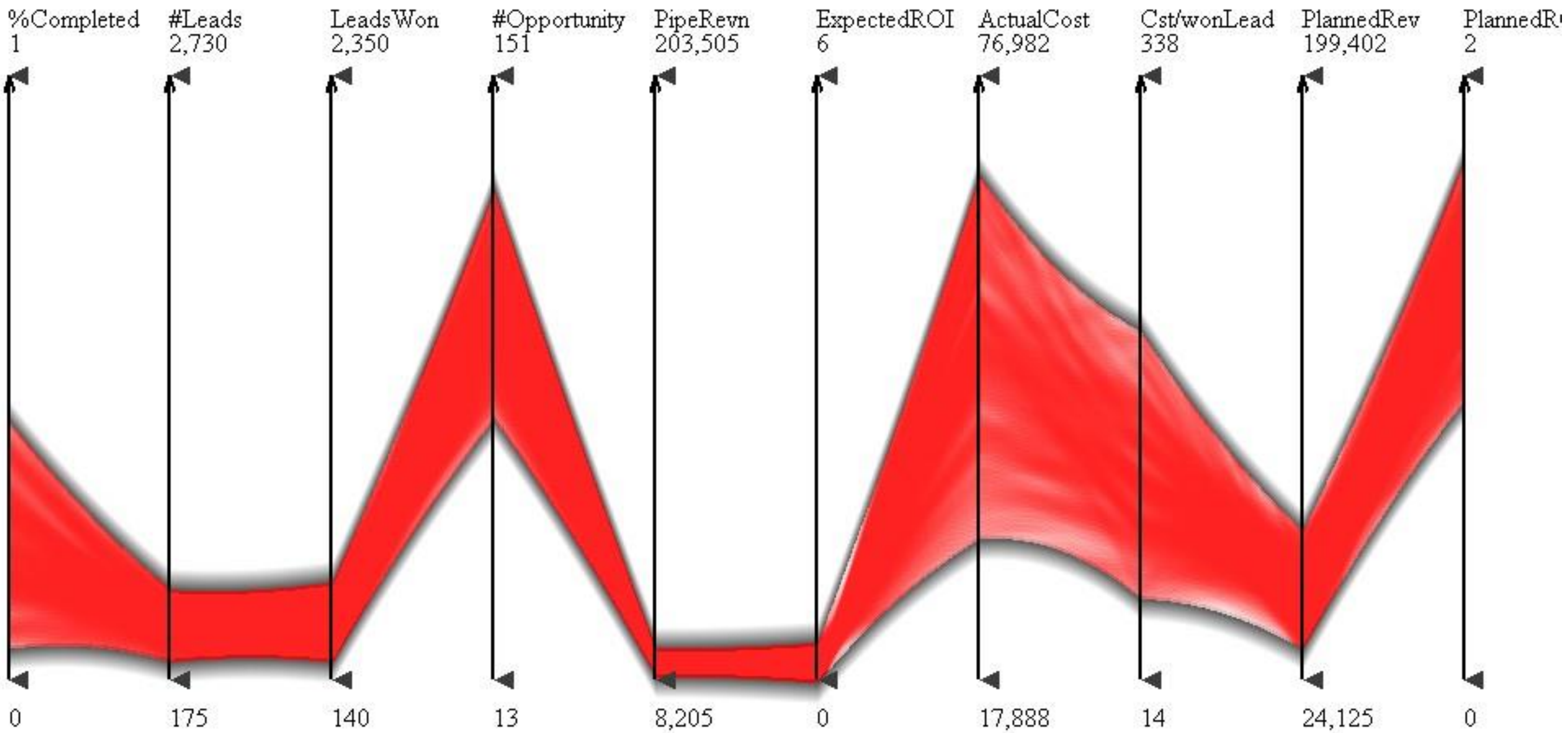
individual polylines

PC With Illustrative Abstraction



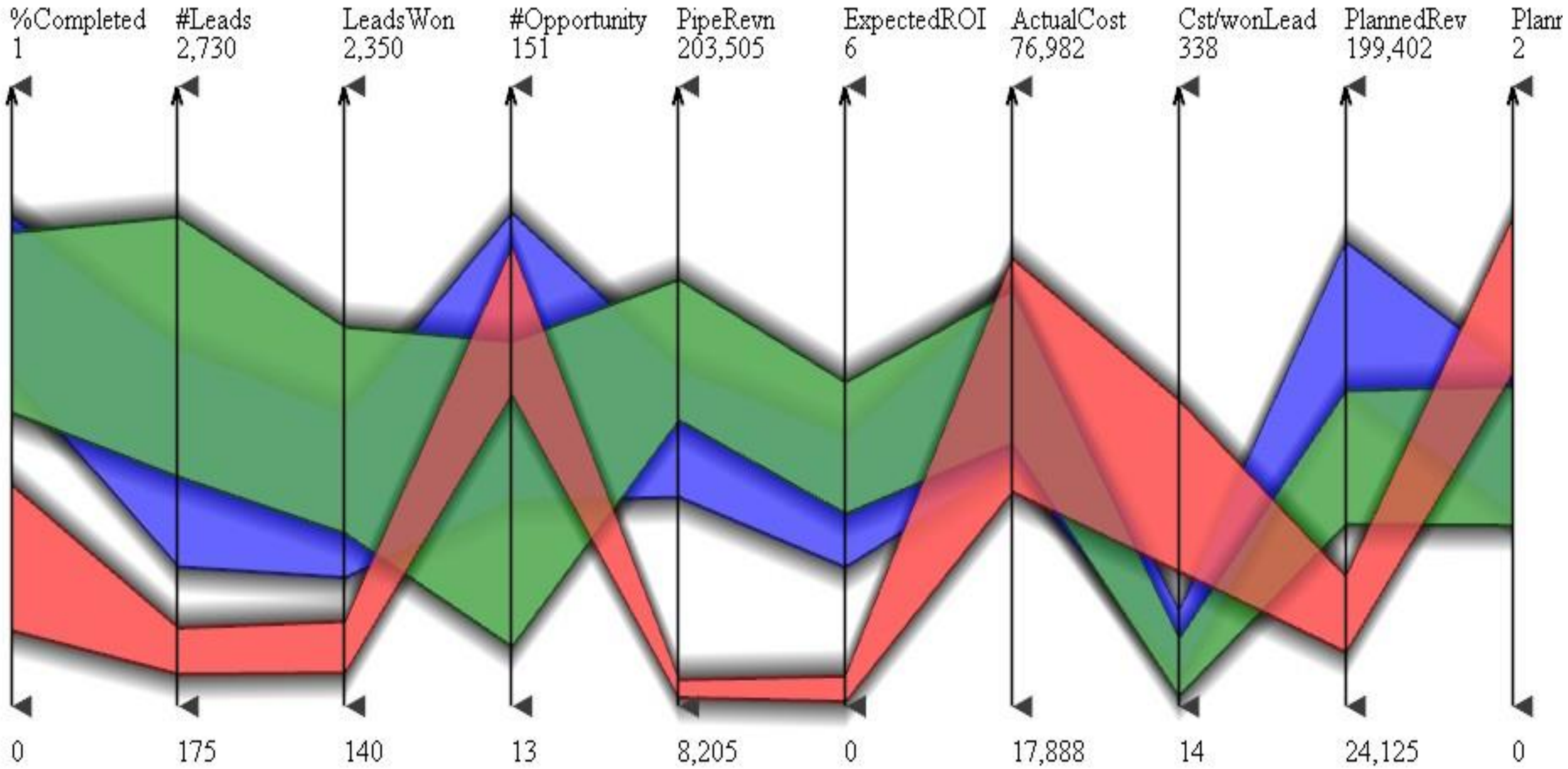
completely abstracted away

PC With Illustrative Abstraction



blended partially

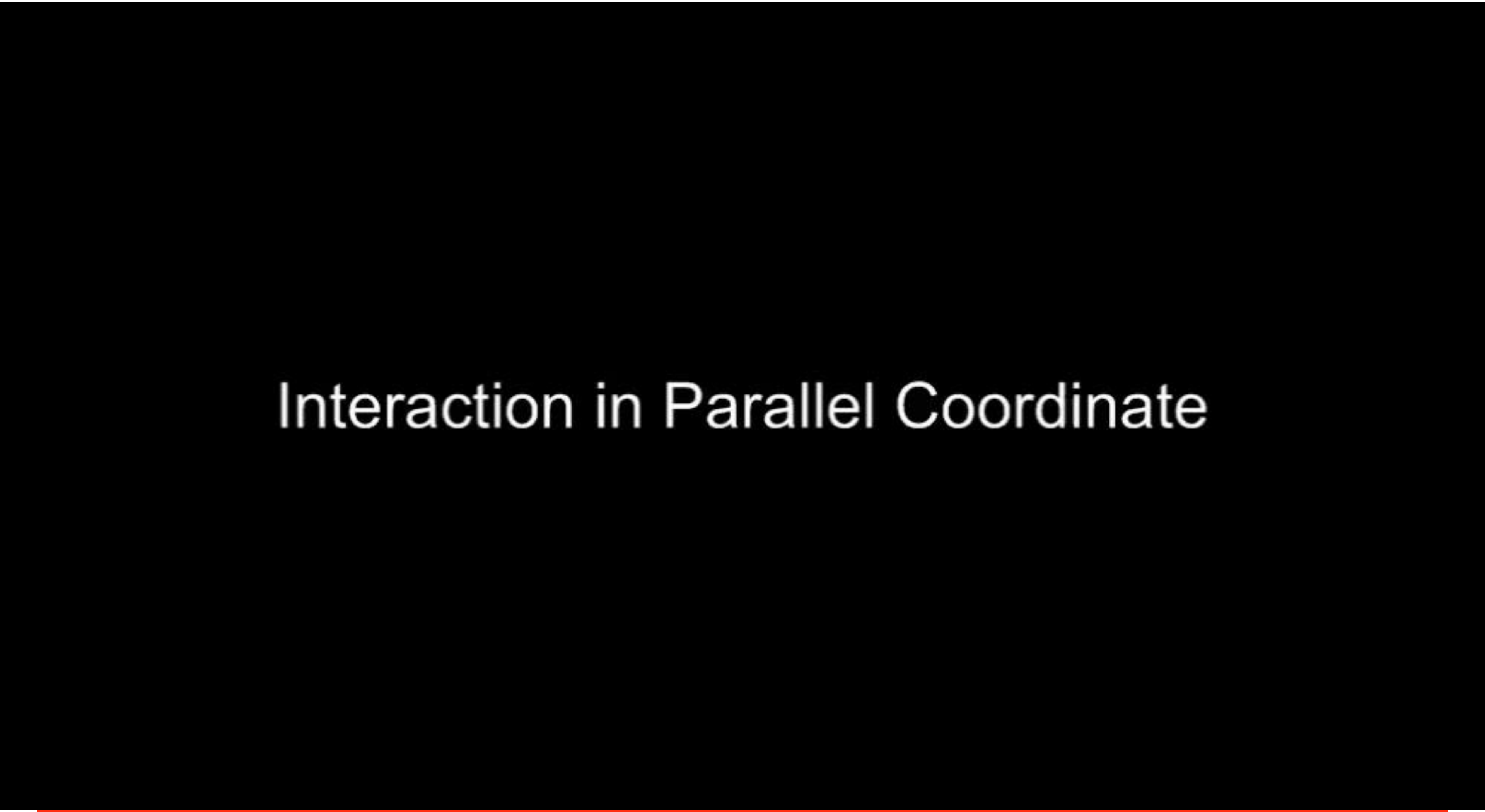
PC With Illustrative Abstraction



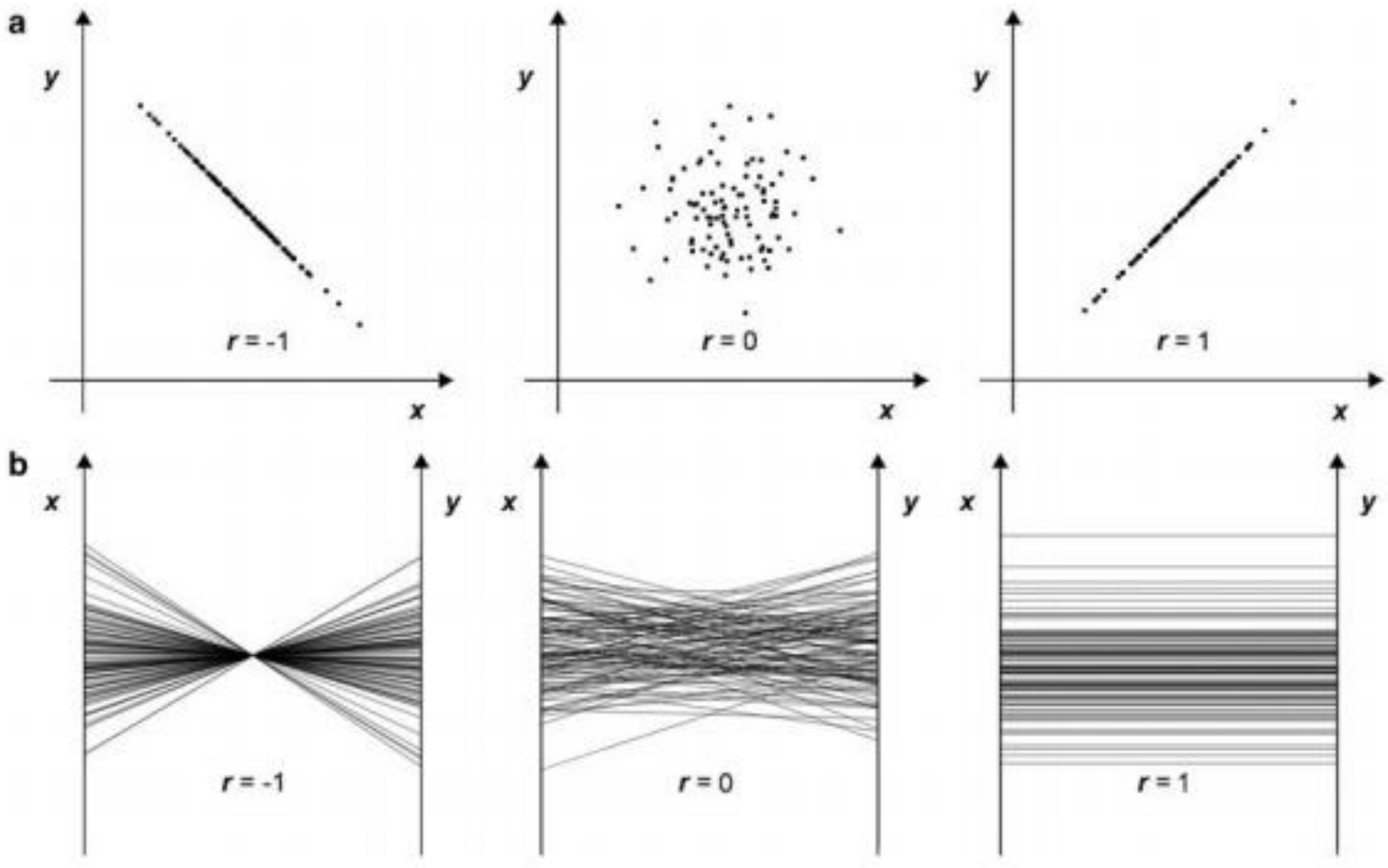
all put together – three clusters

[McDonnell and Mueller, 2008]

Interaction is Key

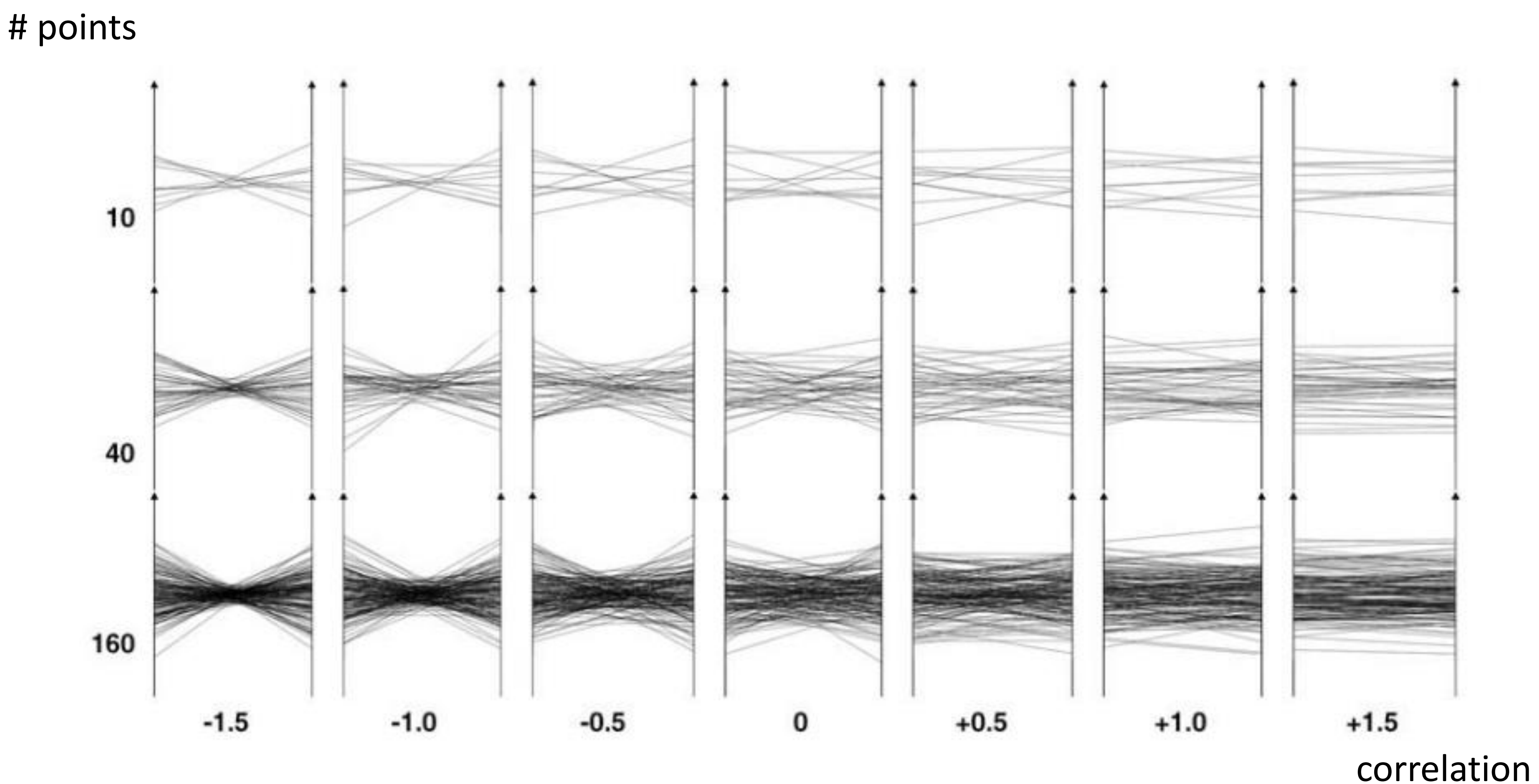


Patterns in Parallel Coordinates

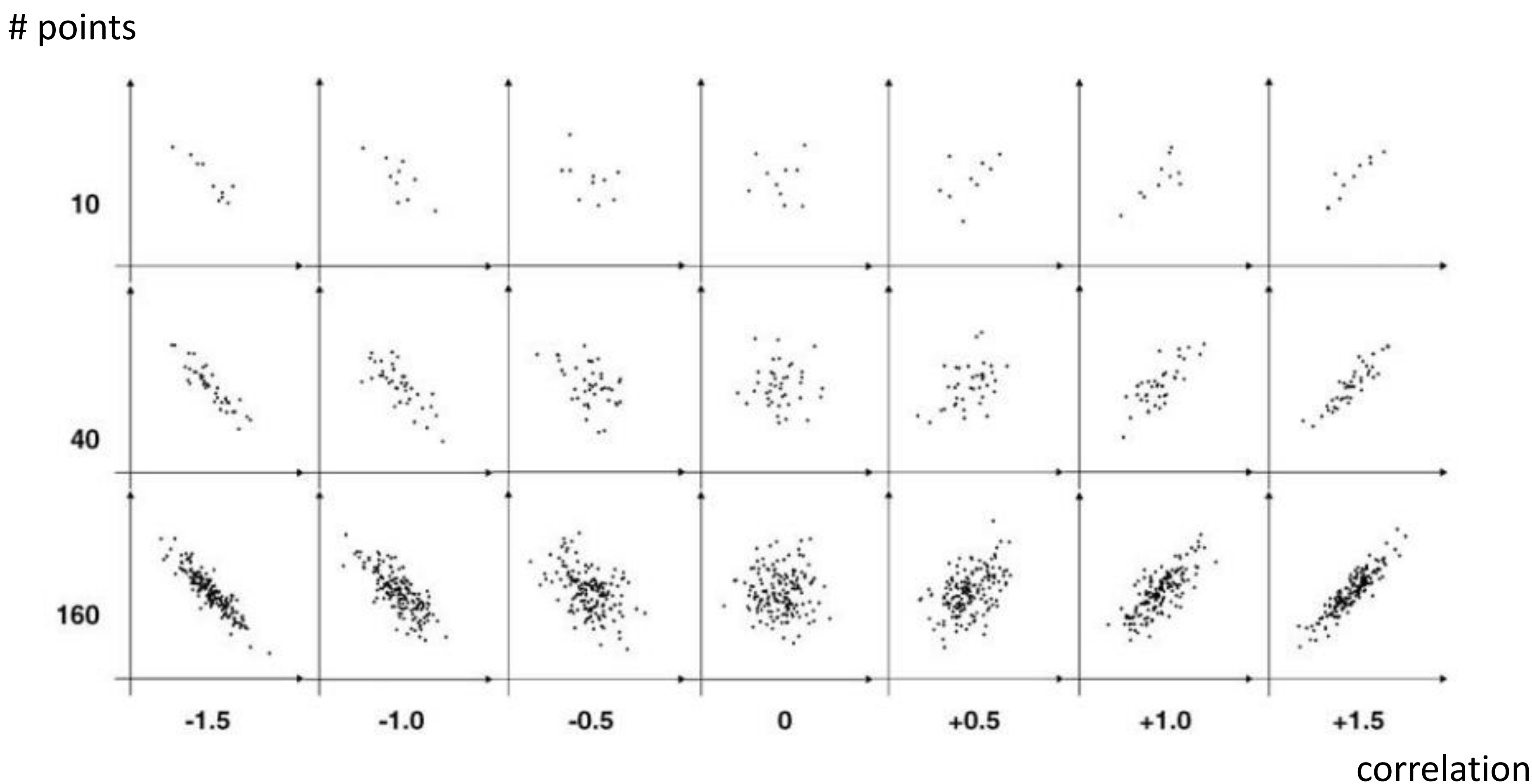


correlation r=-1.0 r=0 r=1.0

Patterns in Parallel Coordinates



Patterns in Scatterplots



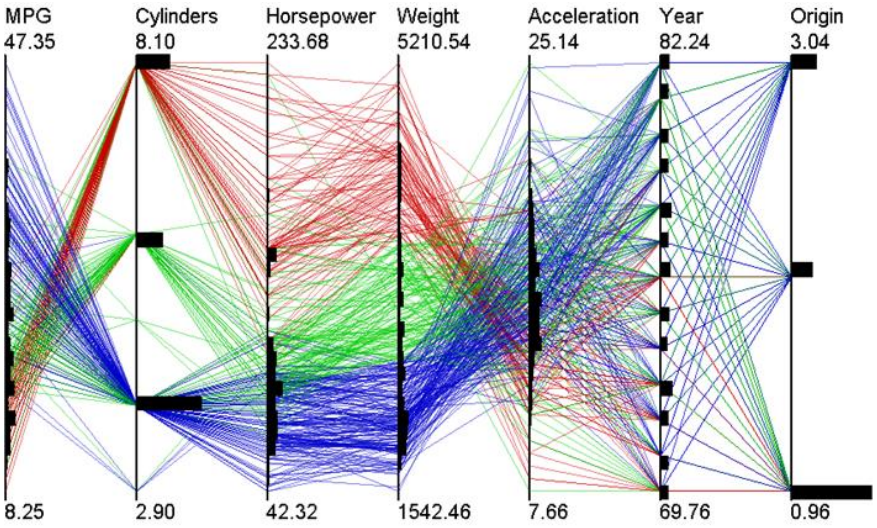
Li et al. found that twice as many correlation levels can be distinguished with scatterplots

Information Visualization Vol. 9, 1, 13 – 30

Axis Reordering Problem

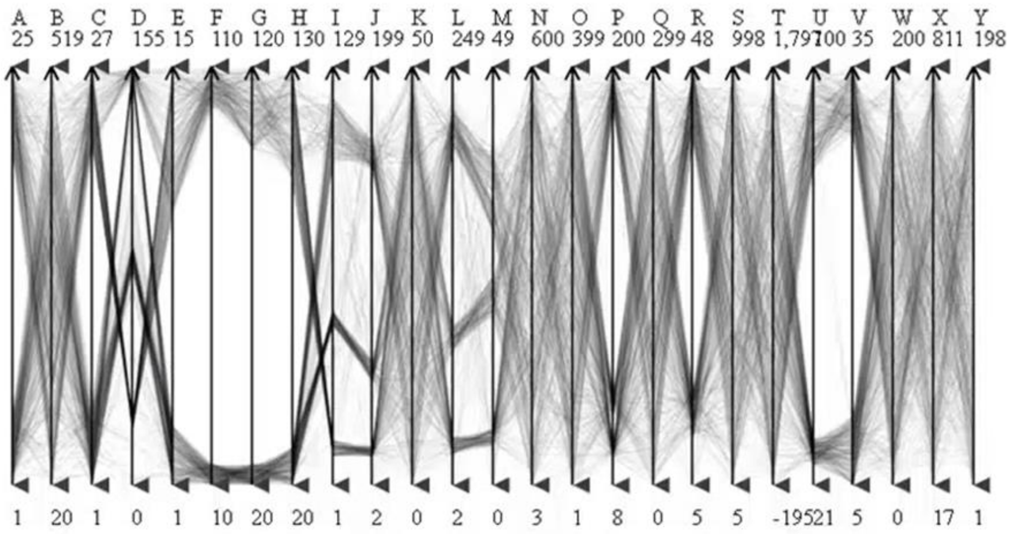
There are n! ways to order the n dimensions

- 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



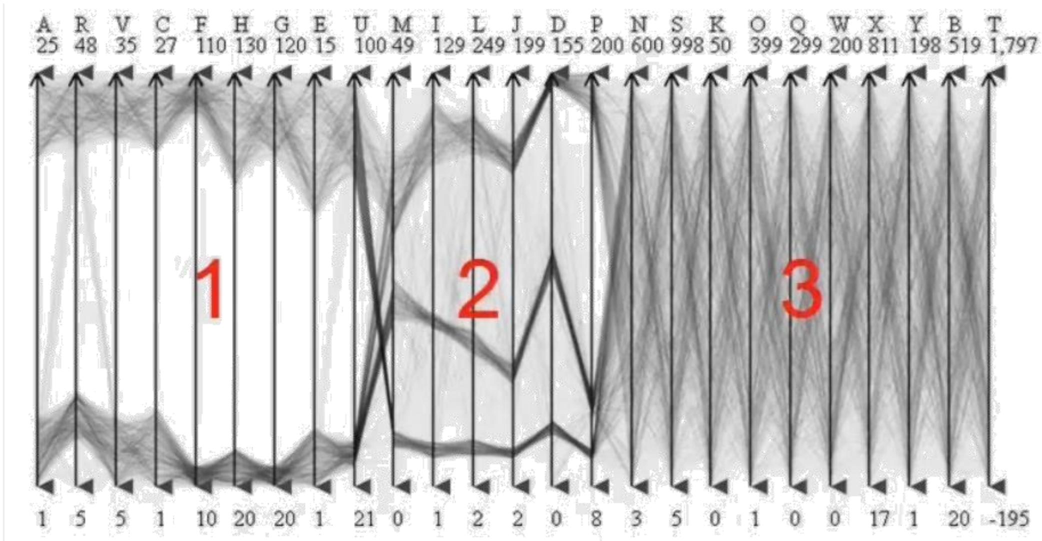
Axis Reordering Problem

The below is not an optimal ordering, why?

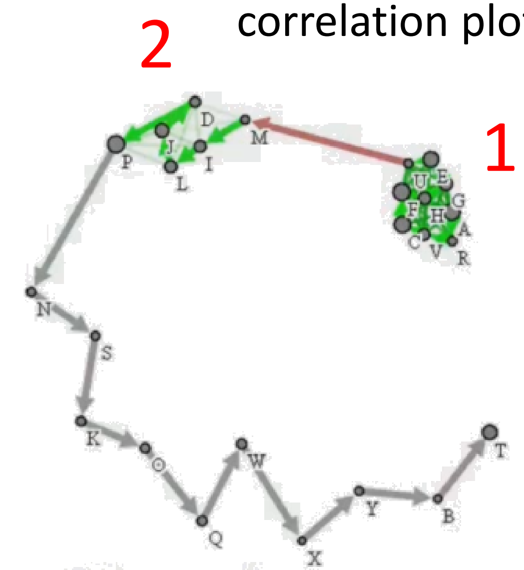


Axis Reordering Problem

This ordering is better, why?



attribute correlation plot



- because it doesn't waste axis pairs on uncorrelated relationships
- only region 3 is uncorrelated
- regions 1 and 2 are subspace clusters

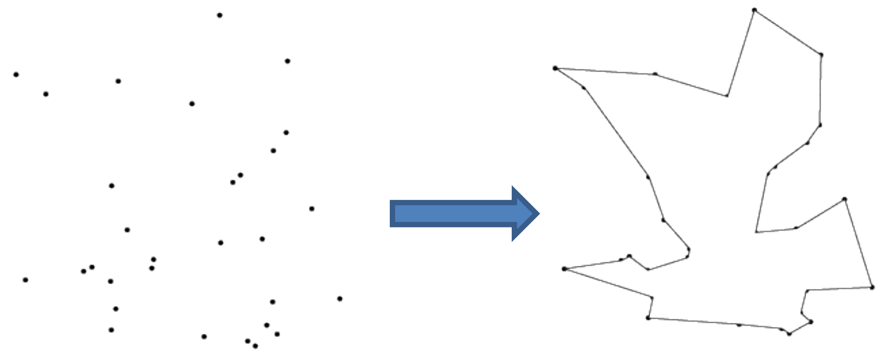
Automatic Axis Ordering

For each axis pair, compute correlation

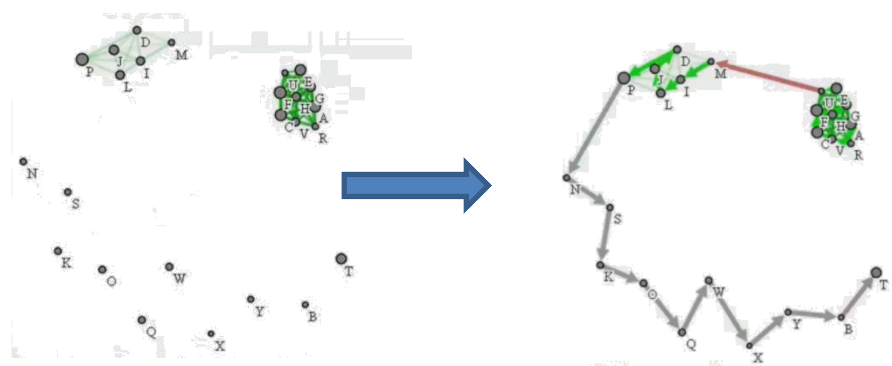
Compute optimal-cost path across all attributes

What algorithm does this?

- Traveling Salesman Solver



Do the same for the correlation plot



Parallel Sets

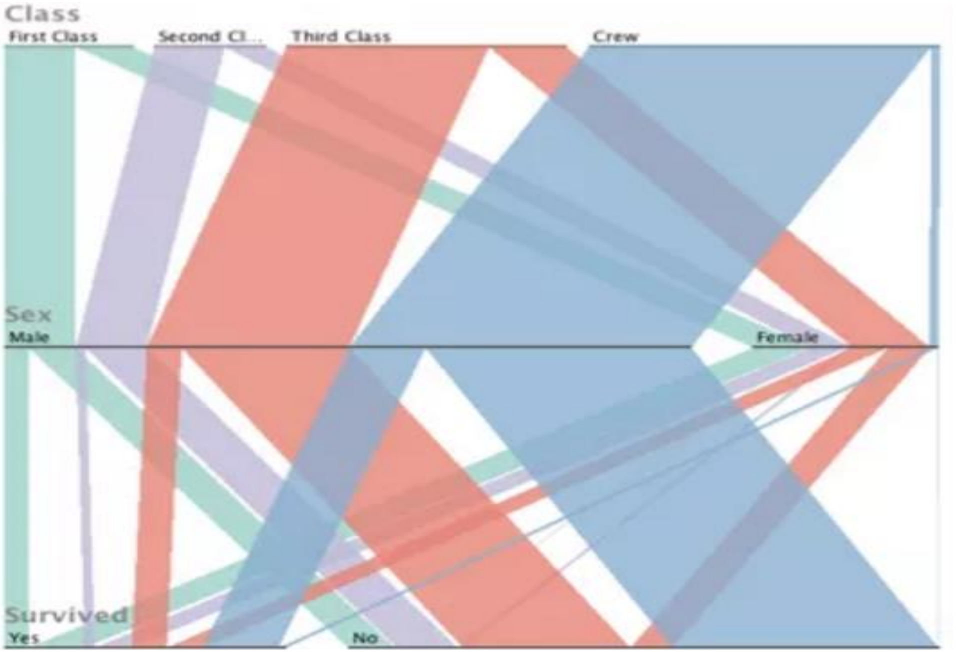
Developed by [Kosara et al. TVCG, 2006]

Parallel coordinates for categorical data

- for example, census and survey data, inventory, etc.
- data that can be summed up in a cross-tabulation

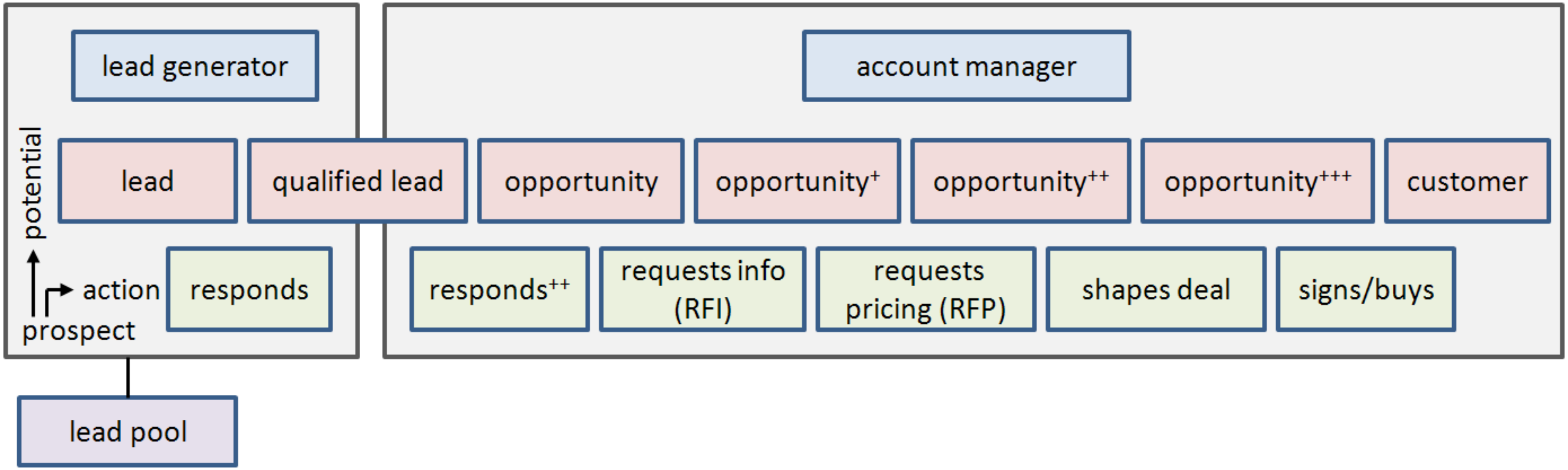
Example

- Titanic dataset
- what can we see here?



Story Telling with Parallel Coordinates

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

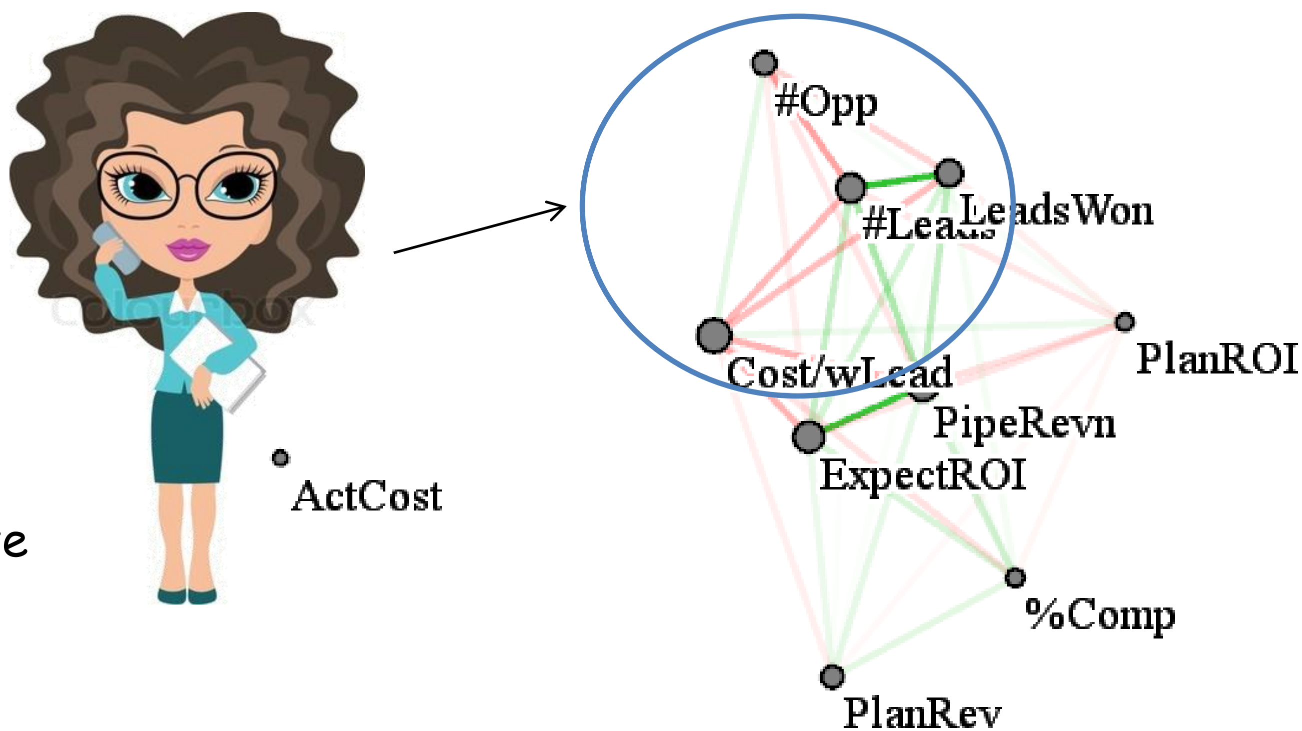
Evidence:

- 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"

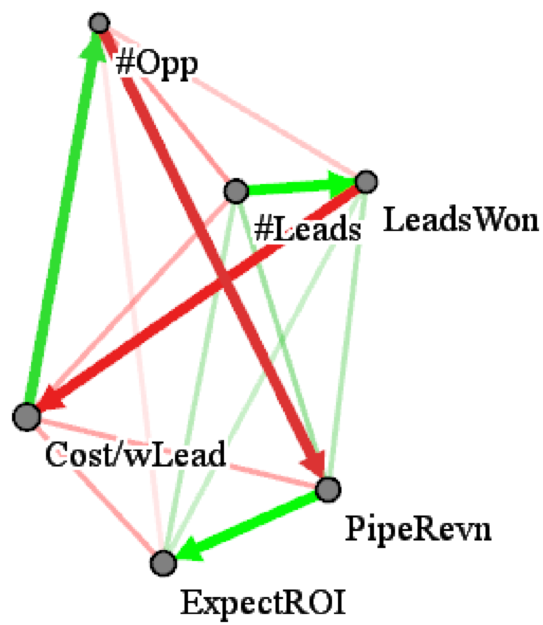


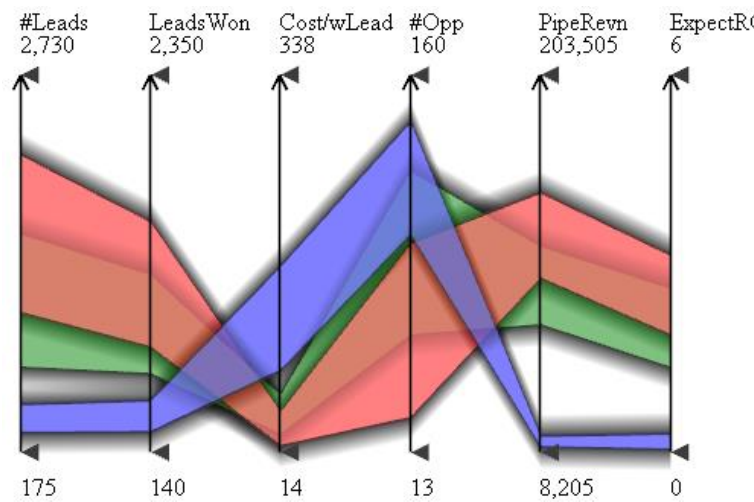
Kate

Kate Designs the Narration

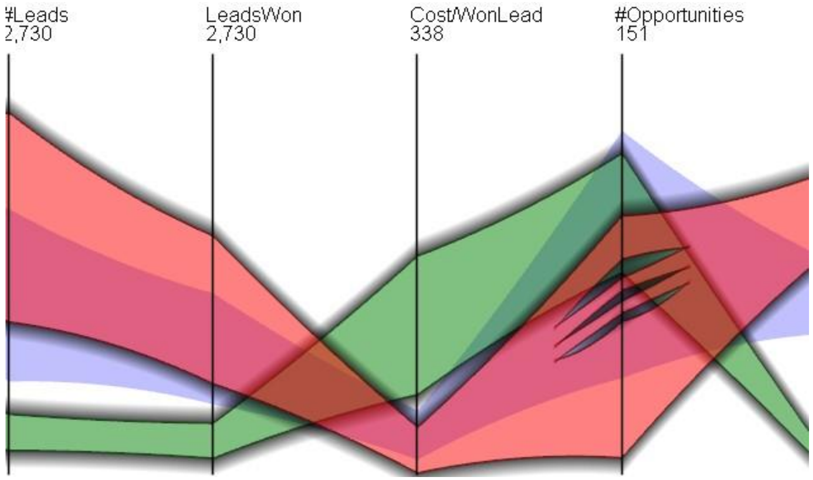
"Let's go and make a revealing route!"

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





Further Insight



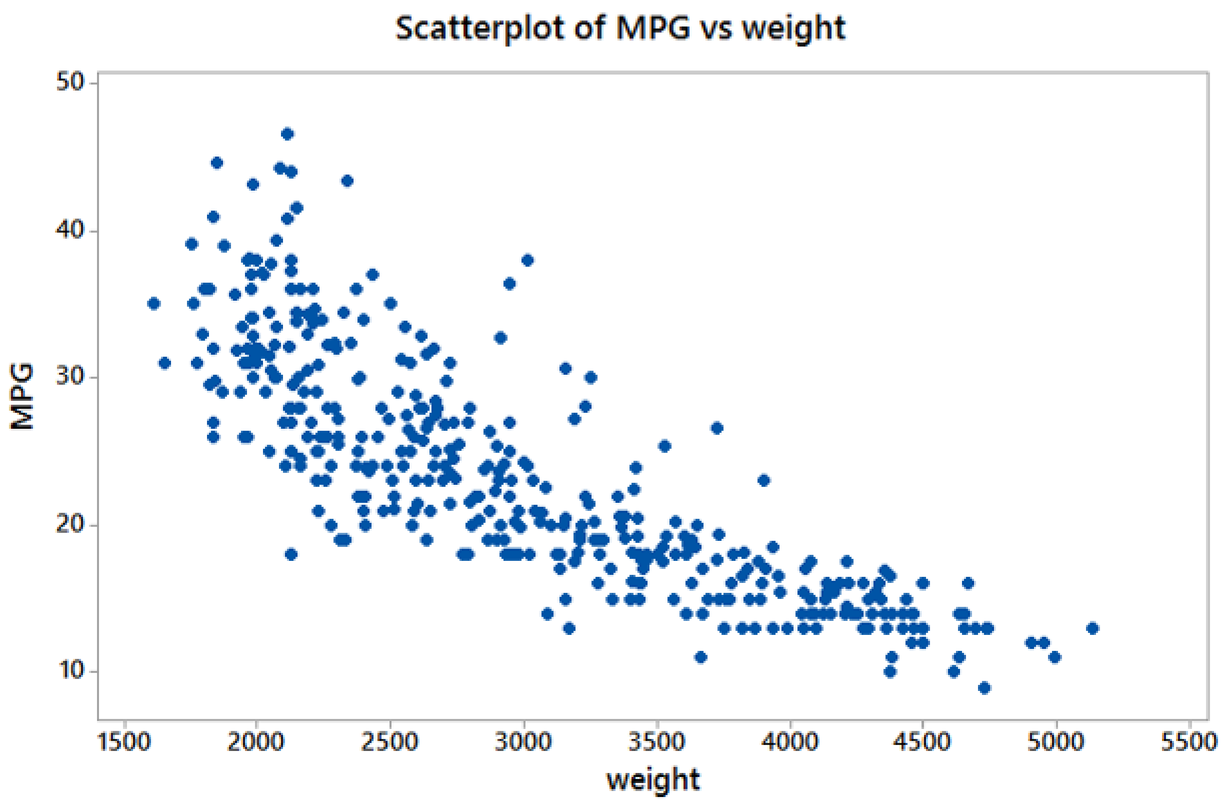
Kate notices something else:

- 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”

Scatterplots

Projection of the data items into a bivariate basis of axes



Projection Operations

How does 2D projection work in practice?

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

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

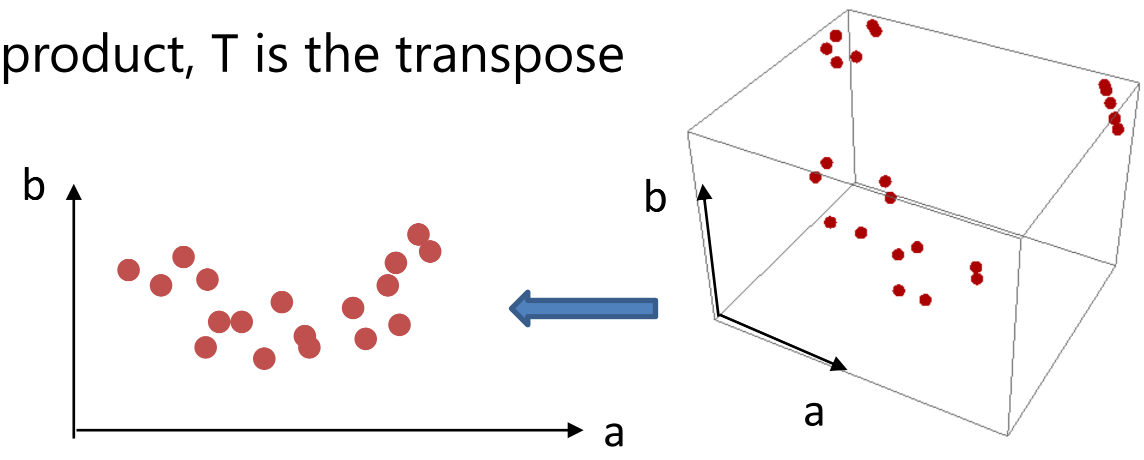
  $b = \{b_1. b_2, b_3, \dots b_N\}$

- a projection $\{x_a, x_b\}$ of x into the 2D basis spanned by {a, b} is:

  $x_a = a \cdot x^T$

  $x_b = b \cdot x^T$

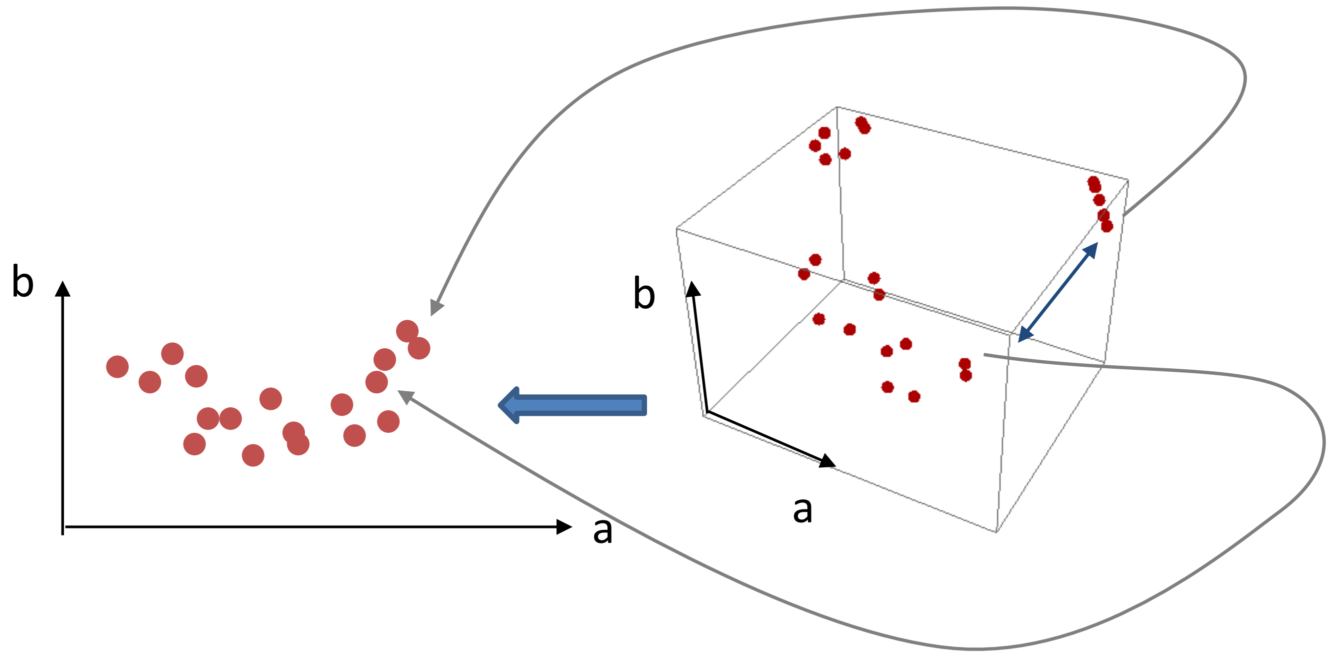
  where · is the dot product, T is the transpose



Projection Ambiguity

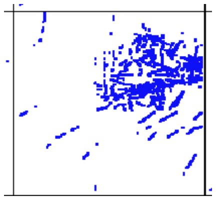
Projection causes inaccuracies

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



Scatterplot For Two Attributes

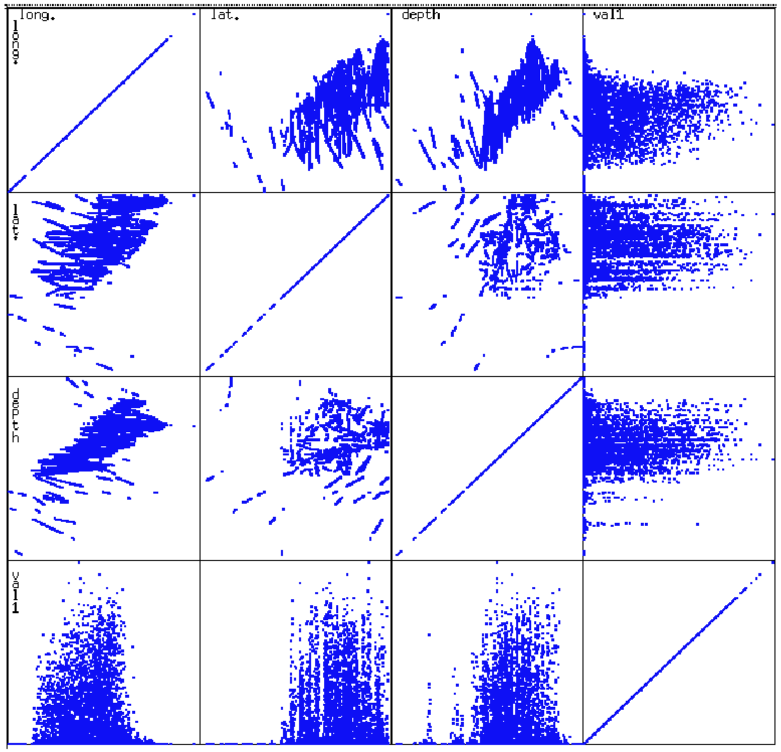
Appropriate for the display of bivariate relationships



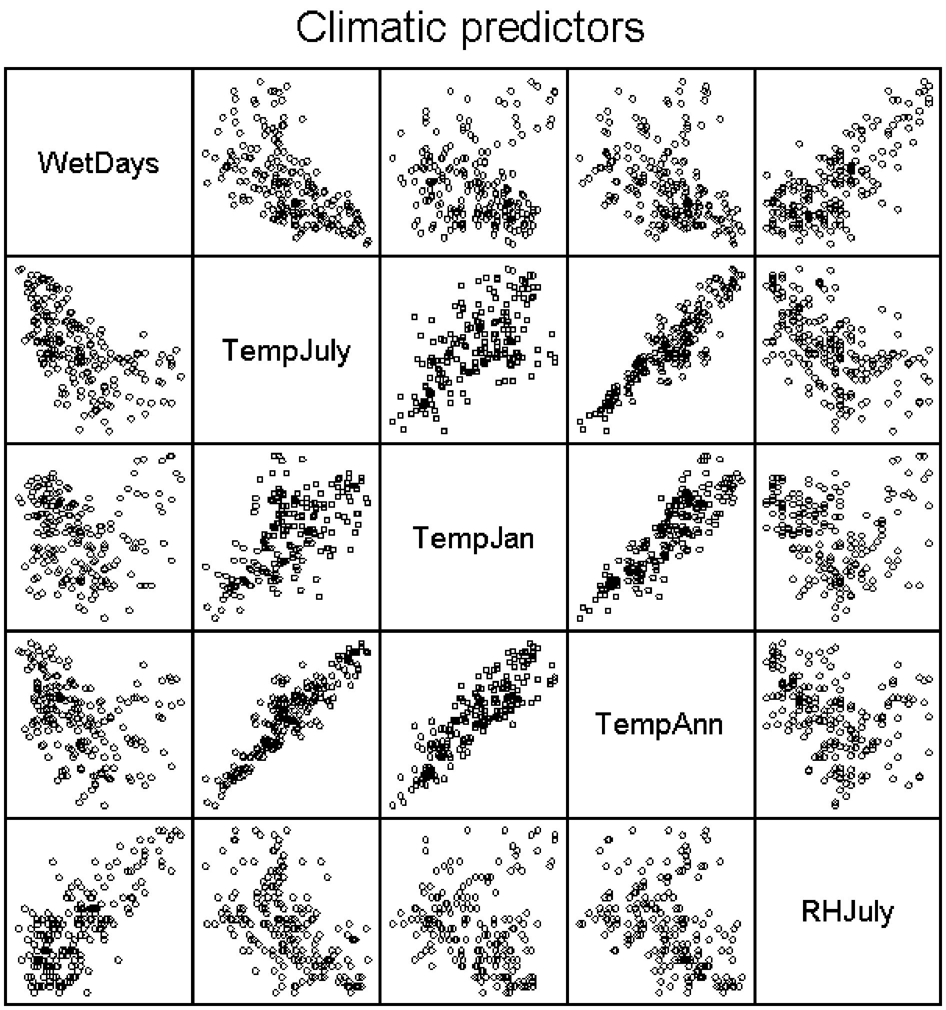
Scatterplot For many Attributes

What to do when there are more than two variables?

- arrange multivariate relationships into scatterplot matrices
- not overly intuitive to perceive multivariate relationships



Scatterplot Matrix (SPLOM)



Scatterplot Matrix

Scatterplot version of parallel coordinates

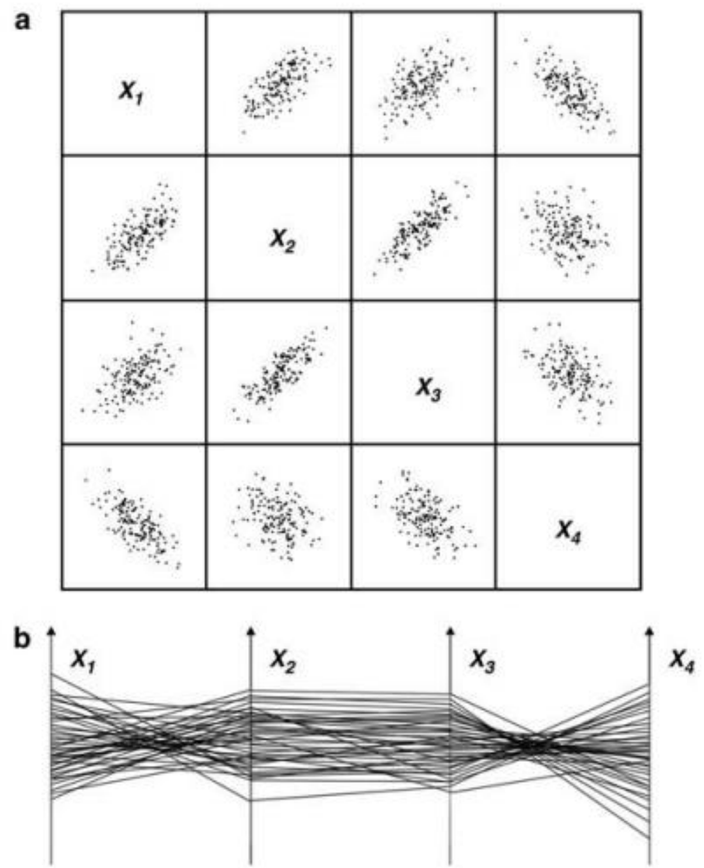
- distributes n(n-1) bivariate relationships over a set of tiles
- for n=4 get 16 tiles
- can use n(n-1)/2 tiles

For even moderately large n:

- there will be too many tiles

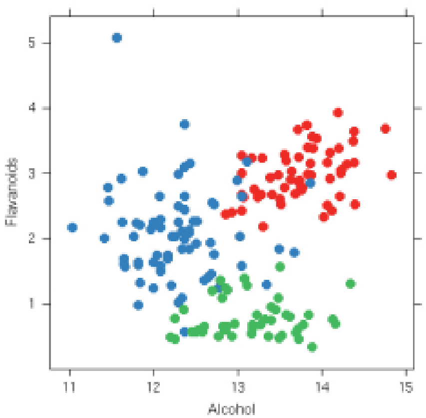
Which plots to select?

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

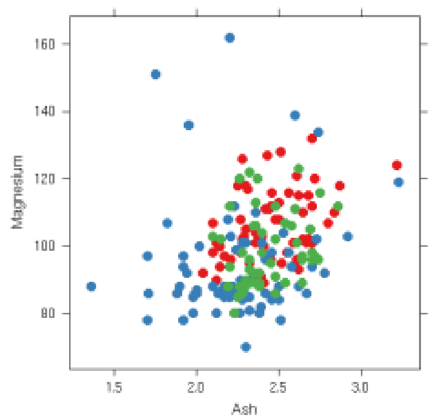


Automated Scatterplot Selection

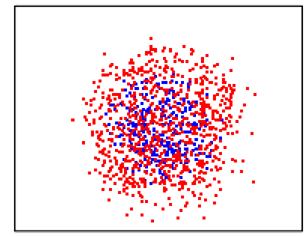
Several metrics, a good one is Distance Consistency (DSC)



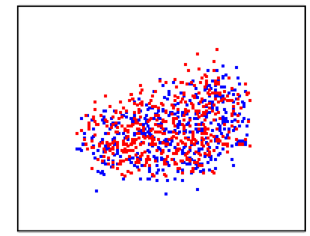
(a) **DSC=90**



(b) **DSC=49**

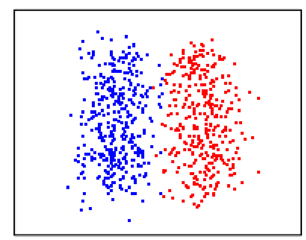


(d) 29

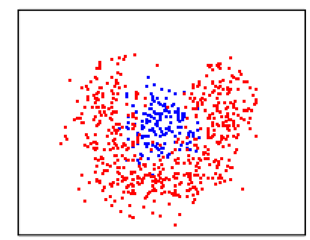


(e) 15

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

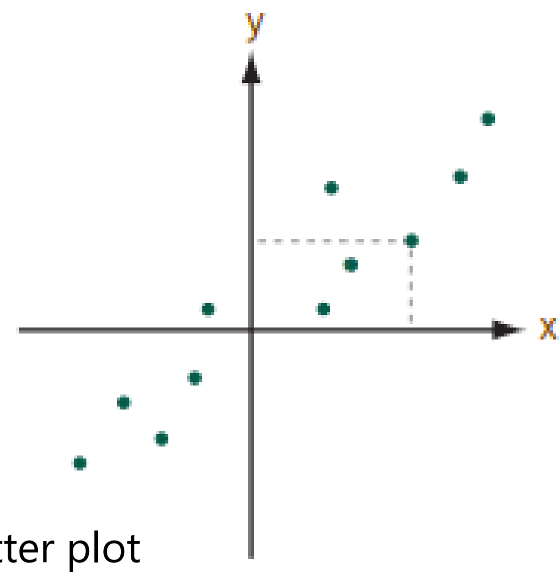
Biplots

Plots data points and dimension axes into a single visualization

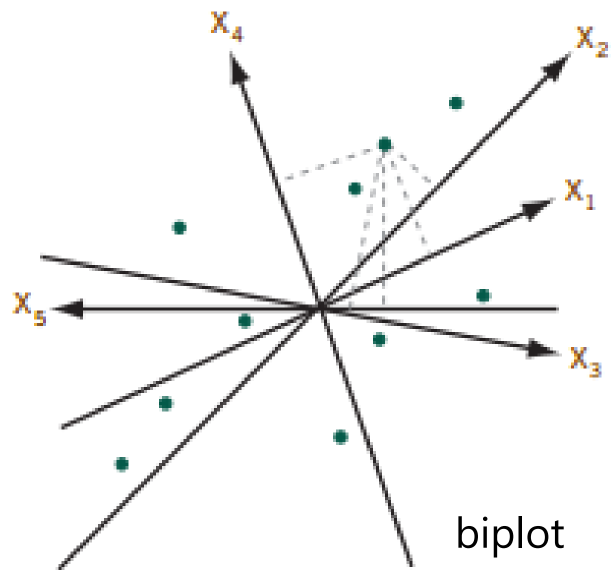
- uses first two PCA vectors as the basis to project into
- find plot coordinates [x] [y]

  for data points: [$PCA_1 \cdot$ data vector] [$PCA_2 \cdot$ data vector]

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



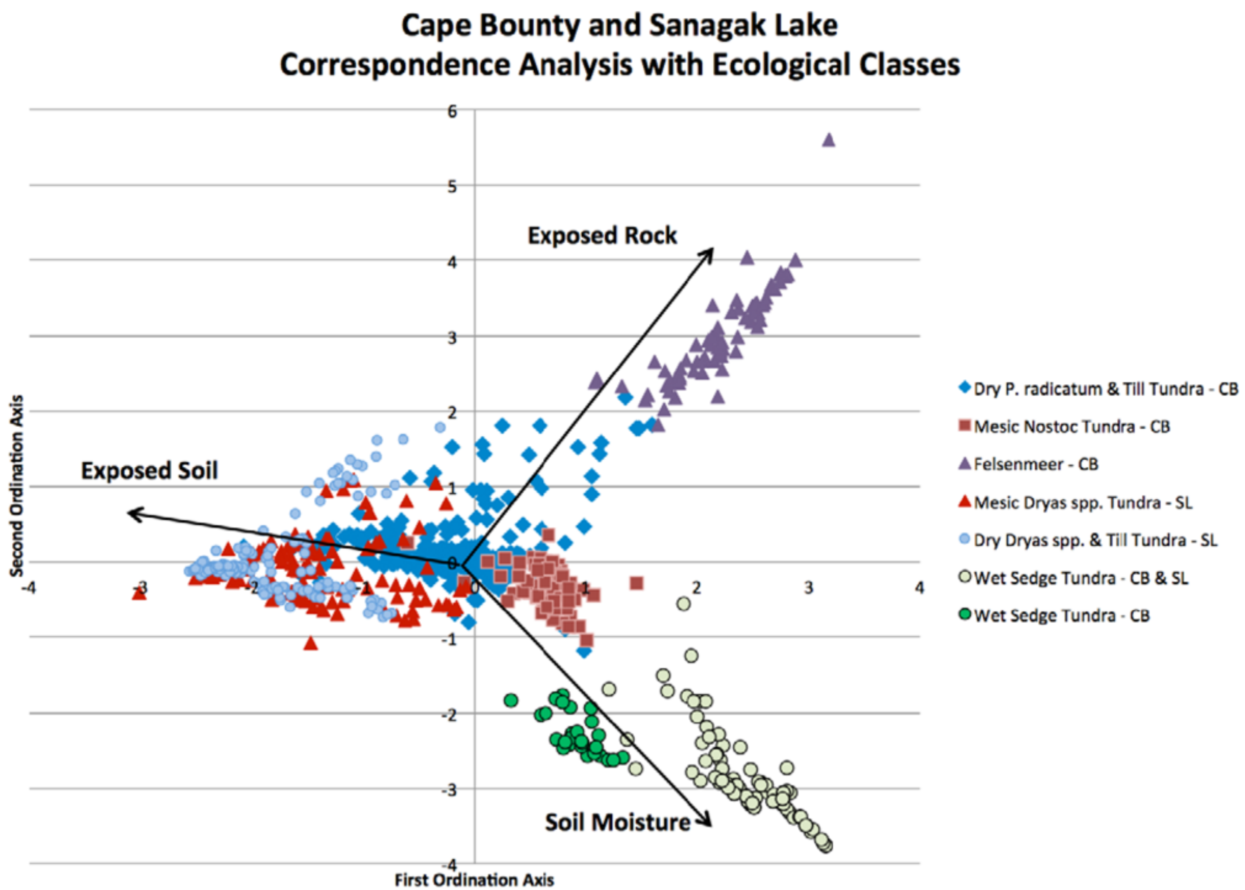
scatter plot



biplot

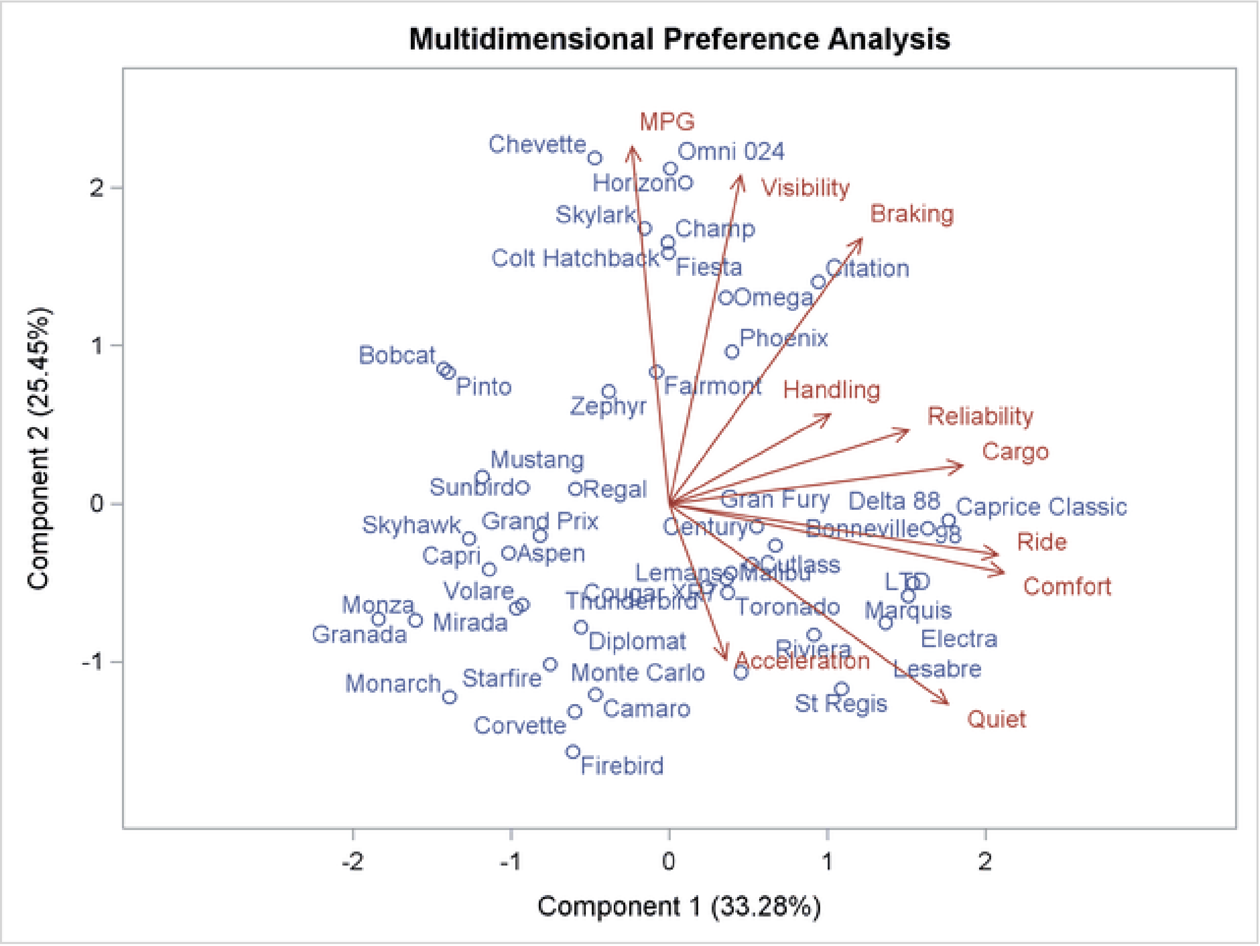
Biplots in Practice

See data distributions into the context of their attributes



Biplots in Practice

See data points into the context of their attributes



Biplots – A Word of caution

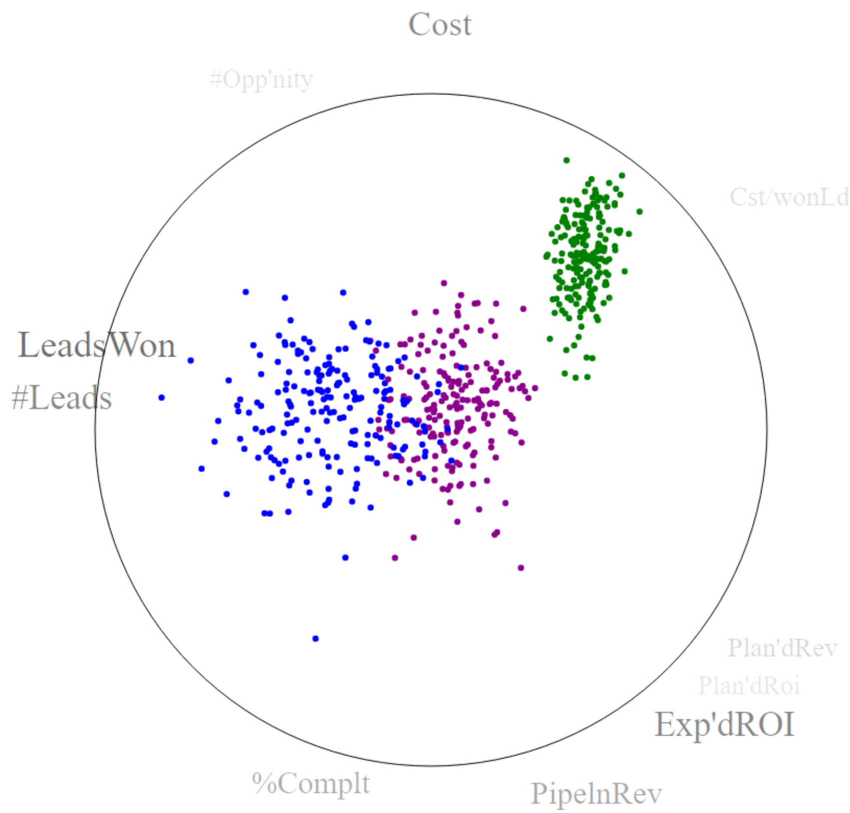
Do be aware that the projections may not be fully accurate

- 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

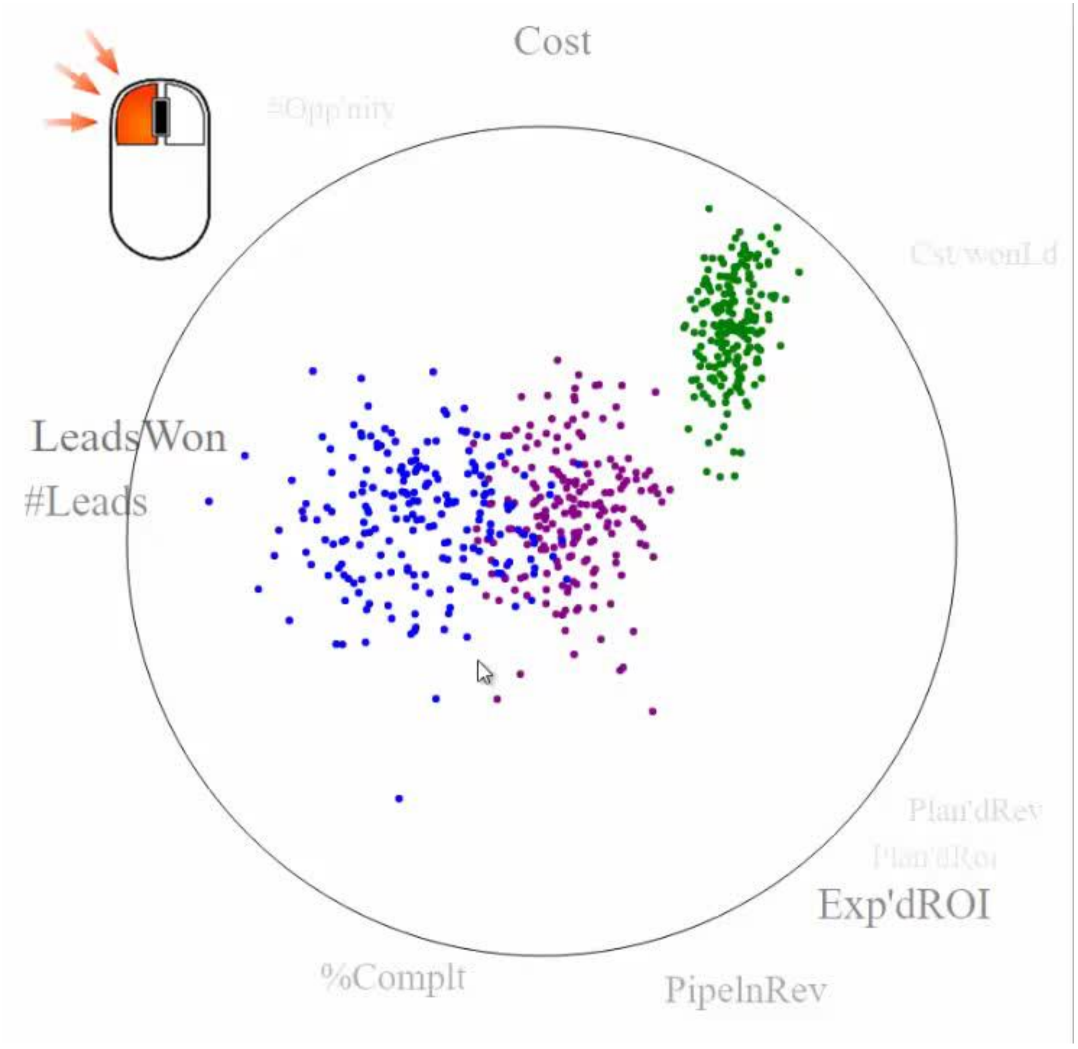
Interactive Biplots

Also called multivariate scatterplot

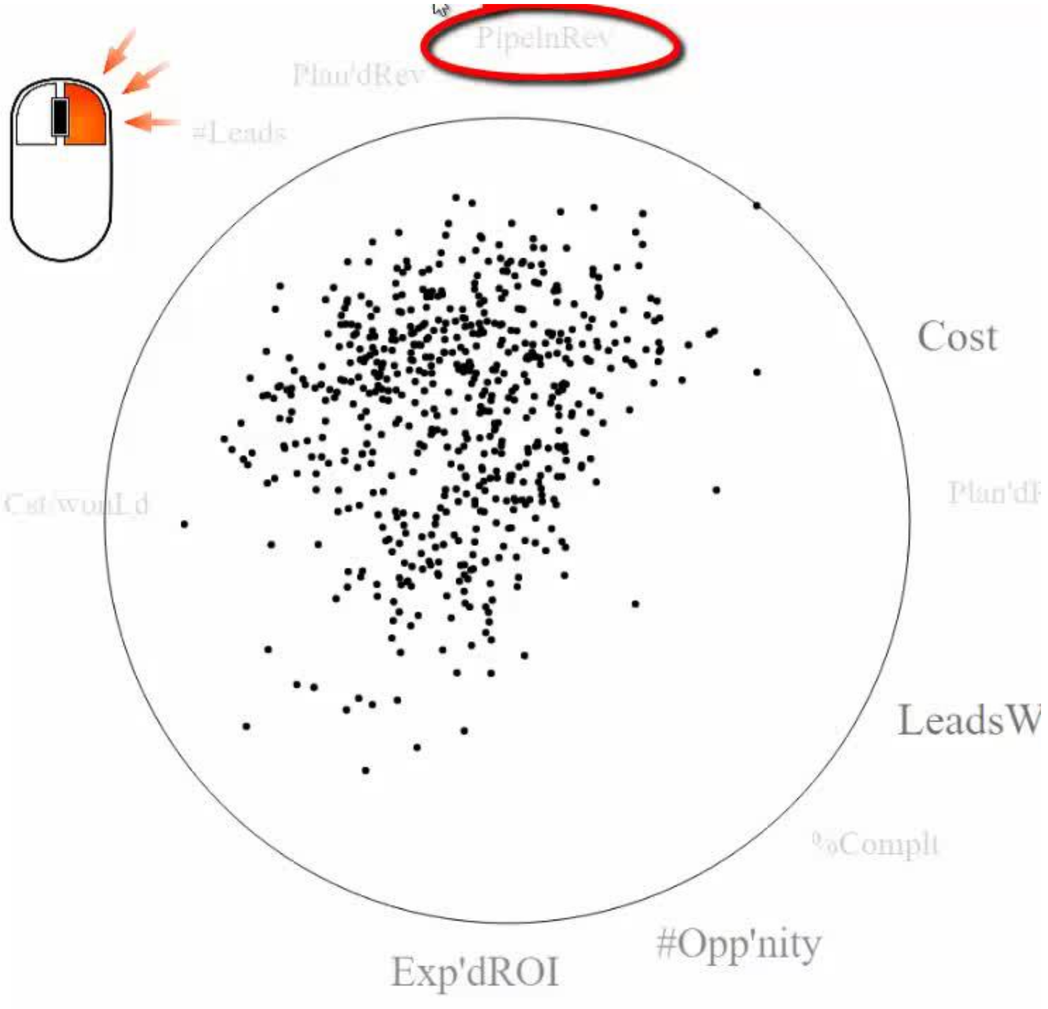
- biplot-axes length vis replaced by graphical design
- less cluttered view
- but there's more to this .....



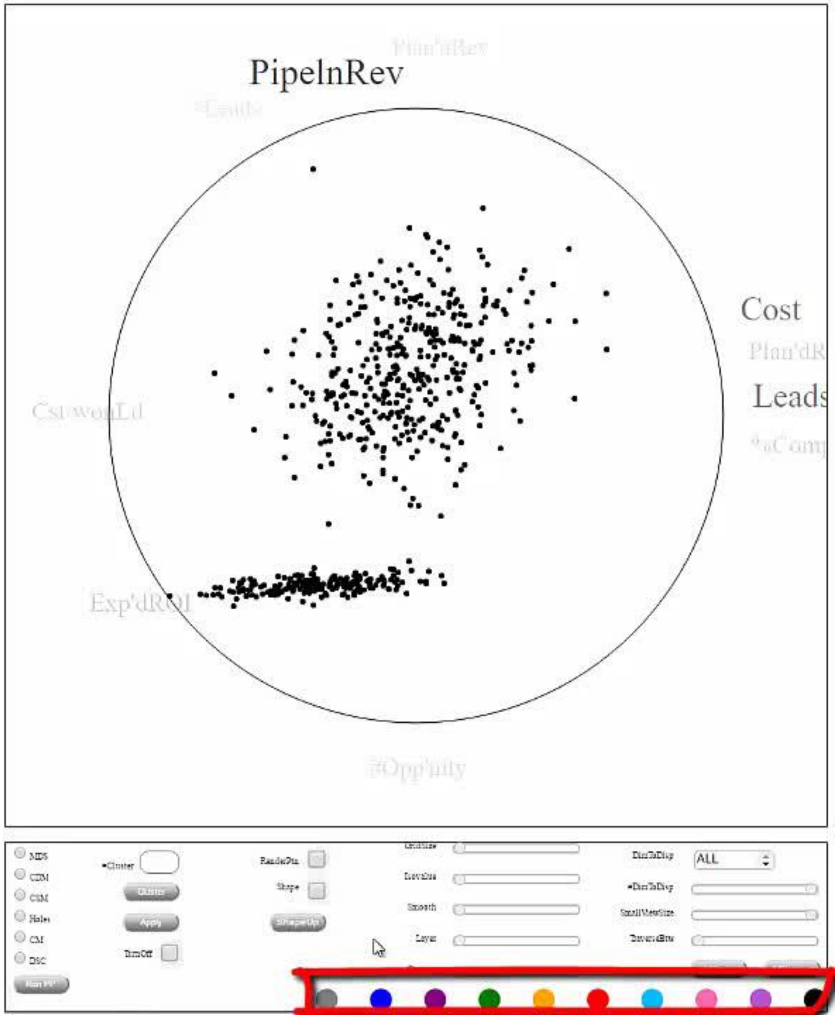
Trackball-Based Cluster Exploration



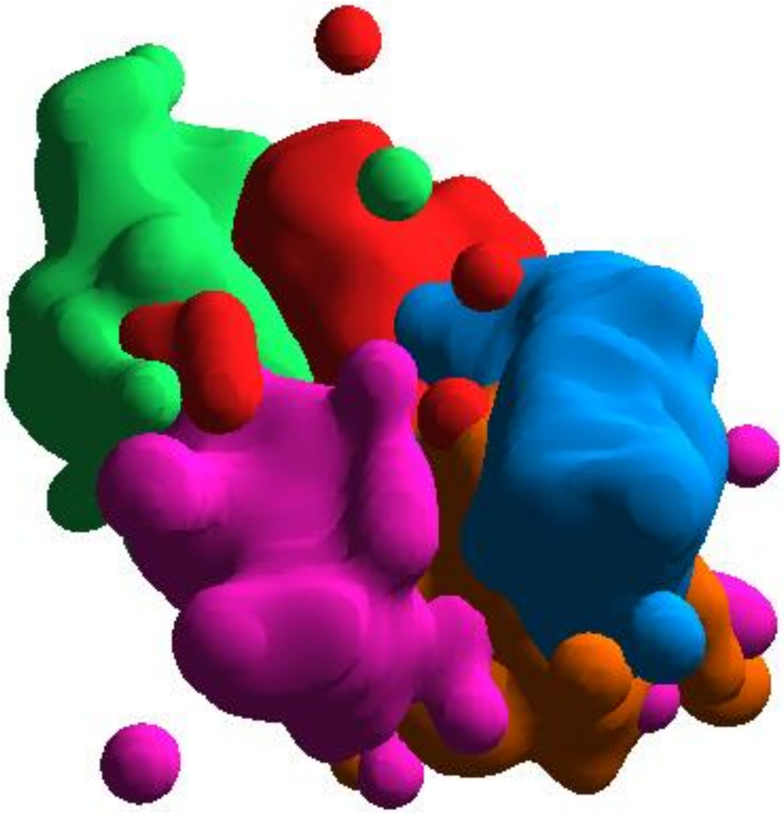
Chase Interesting Clusters – Transition to Adjacent 3D Subspaces



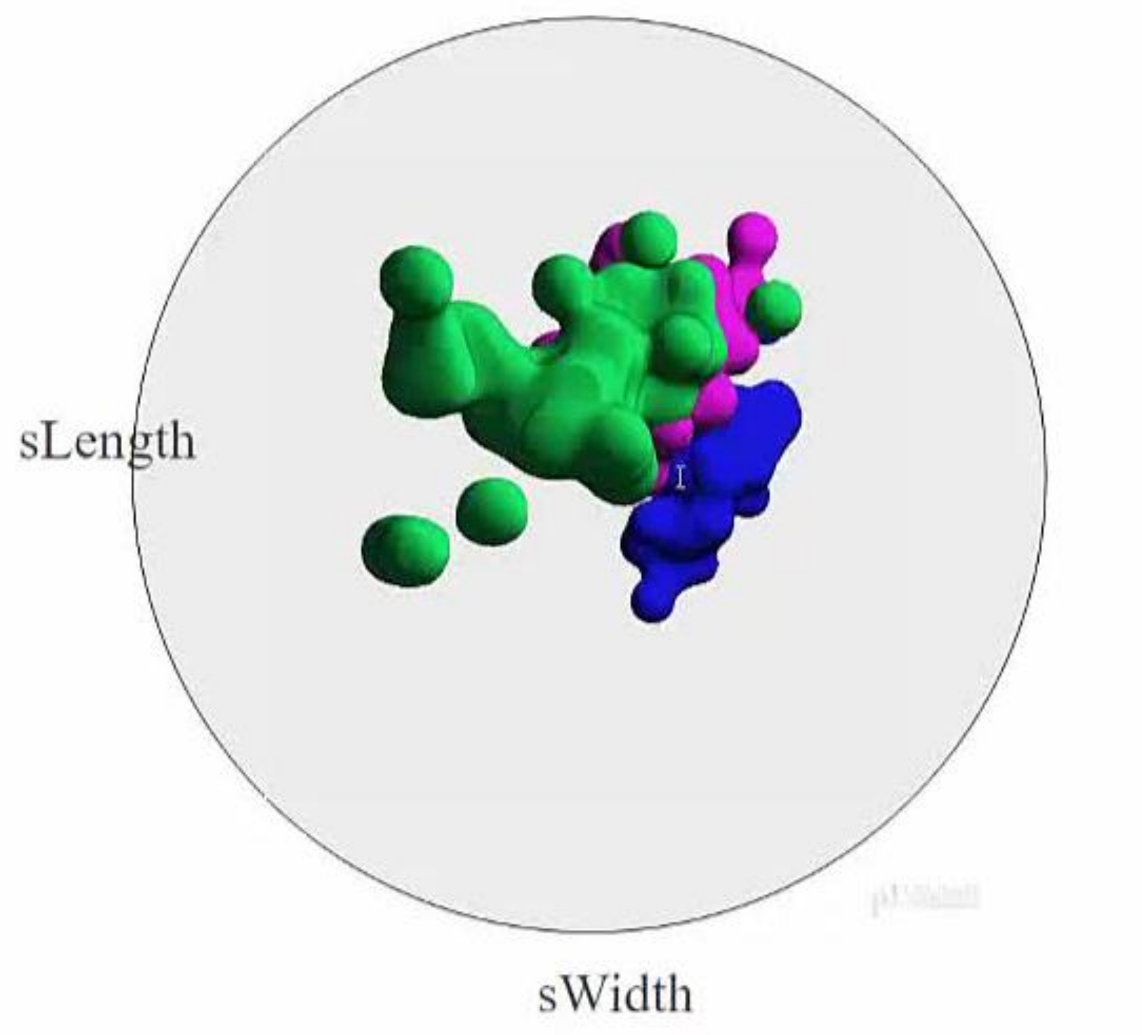
Edit and Annotate Clusters



Clarify Spatial Relationships



Clarify Spatial Relationships



Star Coordinates

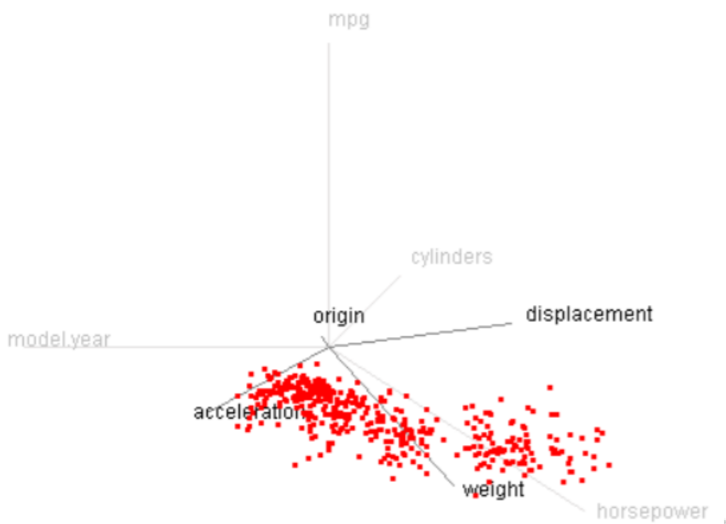
Coordinate system based on axes positioned in a star

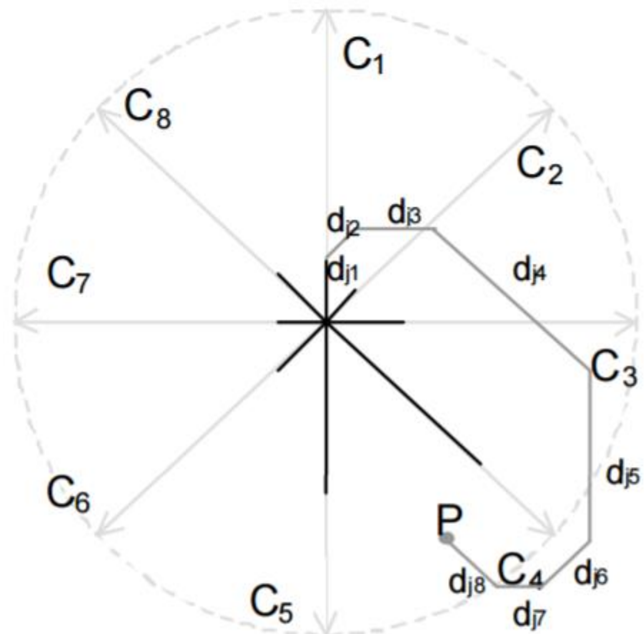
- a point P is vector sum of all axis coordinates

$$P = O + \sum_{i=1}^{m} d_i \vec{c}_i$$

Interactions

- axis rescaling, rotation
- reveal correlations
- resolve plotting ambiguities





[E. Kandogan SIGKDD 2001]

RadViz

Similar to Star Coordinates

- uses a spring model
- difference is normalization by sum of values

$$P = \frac{\sum_{i=1}^{m} d_i \vec{c}_i}{\sum_{i=1}^{m} d_i}$$

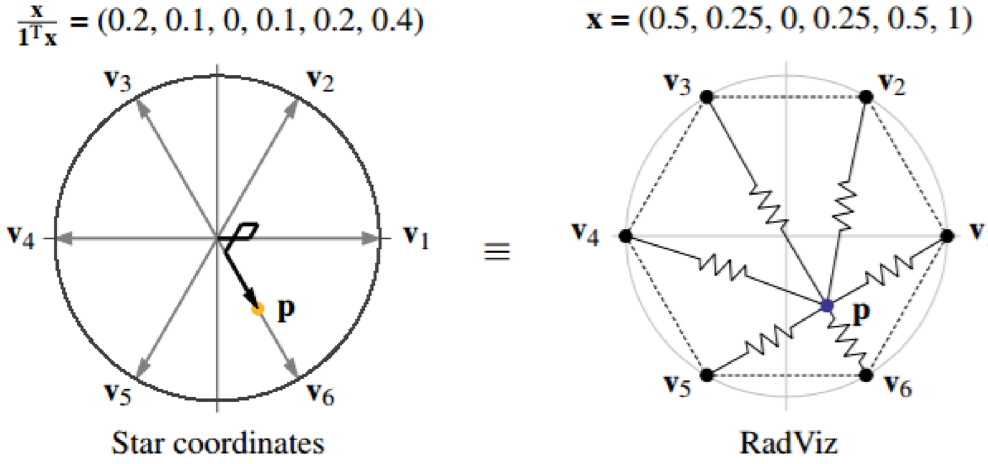
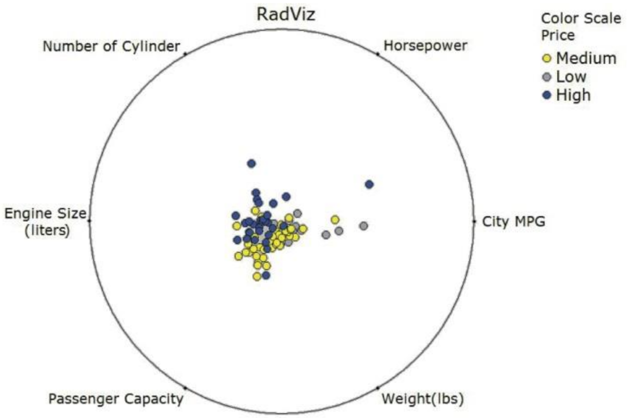
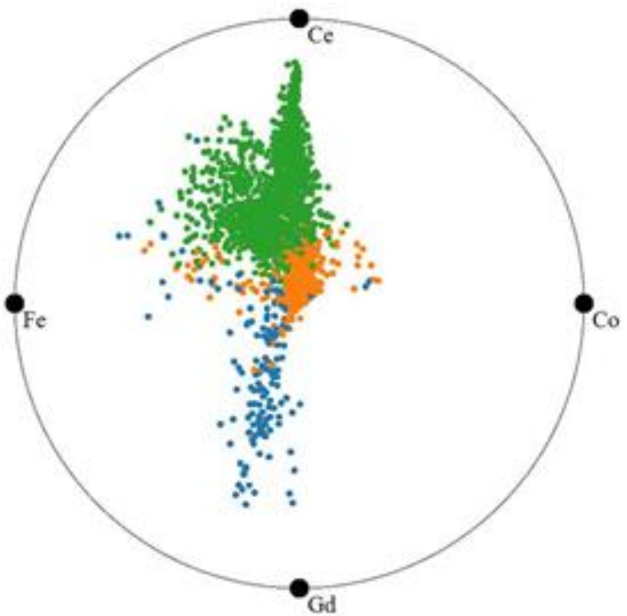


Figure by: Rubio-Sanchez et al. TVCG 2015



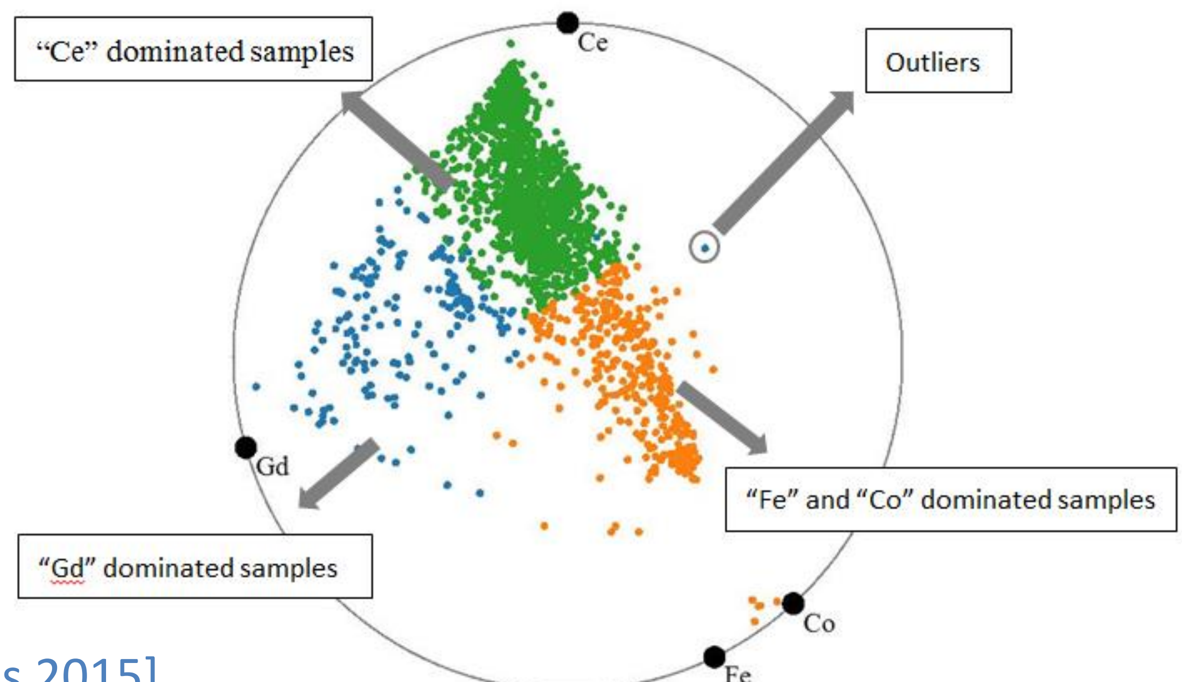
[P. Hoffman et al. VIS 1997]

Optimizing the RadViz Layout



standard RadViz

ɪnt on circle using TSP
iteratively using similarity constraints

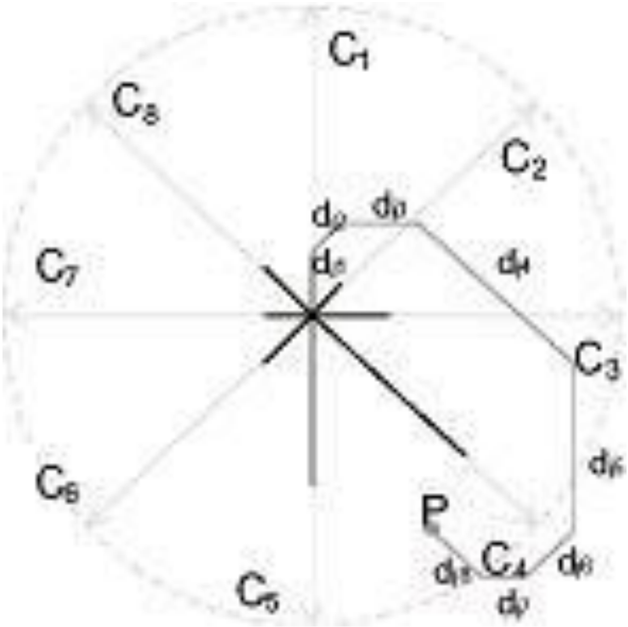


[Cheng and Mueller, Pacific Vis 2015]

Star Coordinates

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

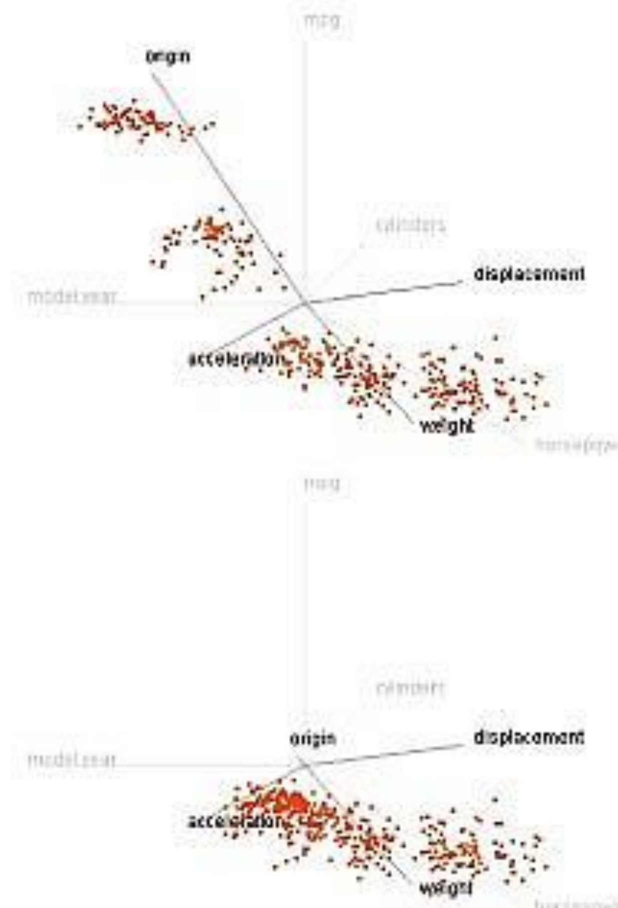
- a point P is plotted as a vector sum of all axis coordinates



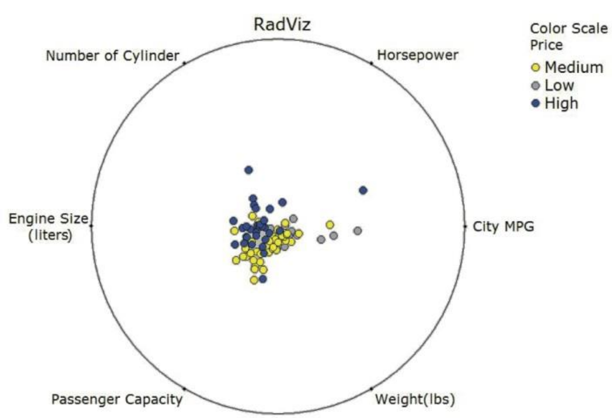
Star Coordinates

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



Commonalities

All of these 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
- these are also called *lower-dimensional embeddings*
- very popular is MDS (Multi-dimensional scaling)

Multidimensional Scaling (MDS)

MDS is for irregular structures

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

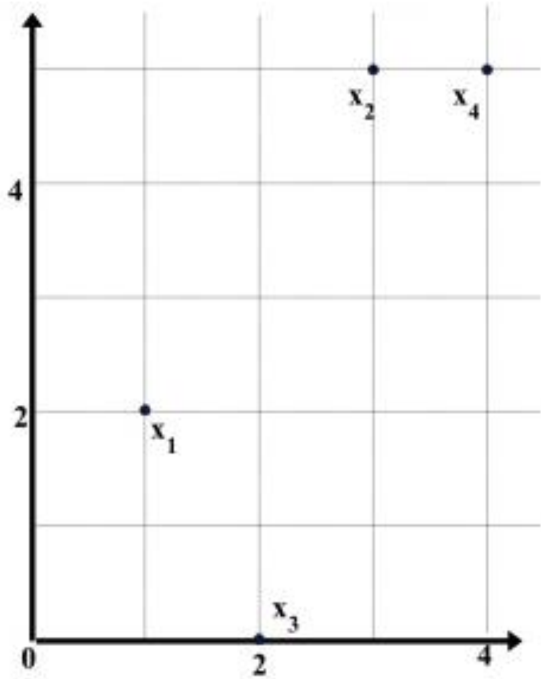
Maps the distances between observations from N-D into low-D (say 2D)

- 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

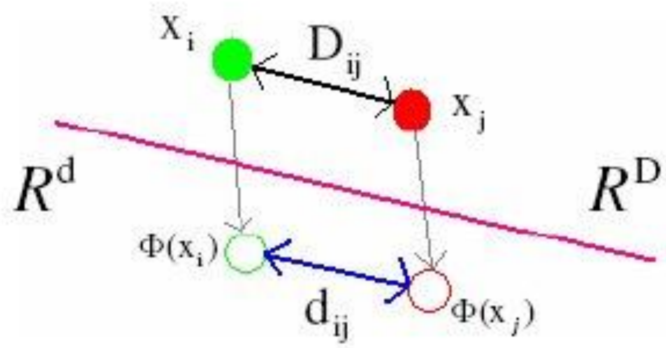
	Chicago	Raleigh	Boston	Seattle	S.F.	Austin	Orlando
Chicago	0						
Raleigh	641	0					
Boston	851	608	0				
Seattle	1733	2363	2488	0			
S.F.	1855	2406	2696	684	0		
Austin	972	1167	1691	1764	1495	0	
Orlando	994	520	1105	2565	2458	1015	0

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):

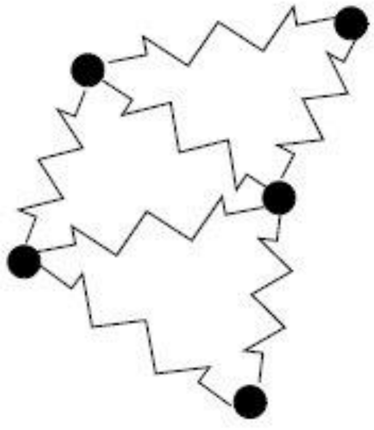
- iterative procedure to find a good configuration of image points
  - 1) Initialization
    → Begin with some (arbitrary) initial configuration
  - 2) Alter the image points and try to find a configuration of points that minimizes the following sum-of-squares error function:

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

Force-Directed Algorithm

Spring-like system

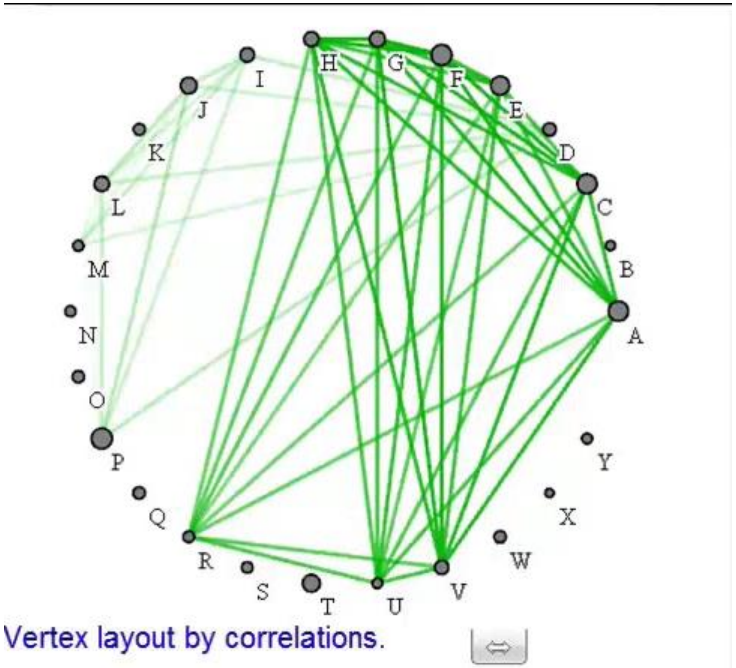
- insert springs within each node
- the length of the spring encodes the desired node distance
- start at an initial configuration
- iteratively move nodes until an energy minimum is reached



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.

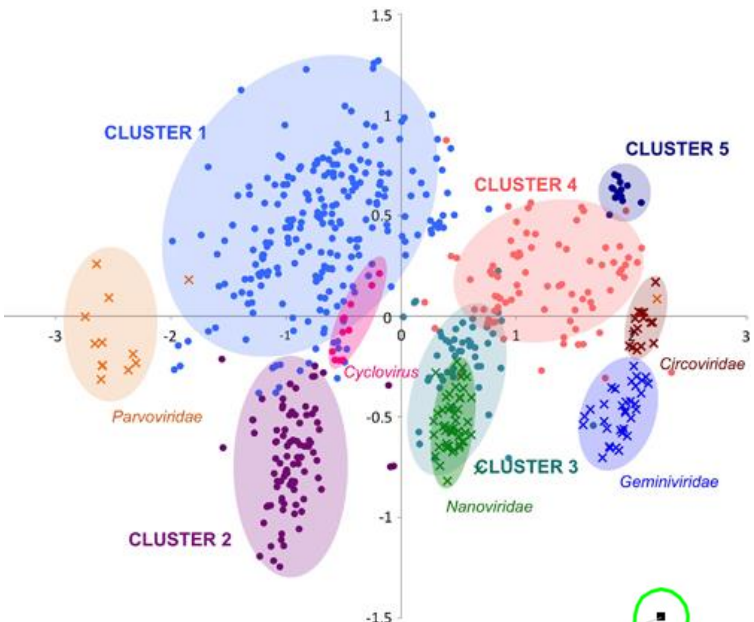
Uses of MDS

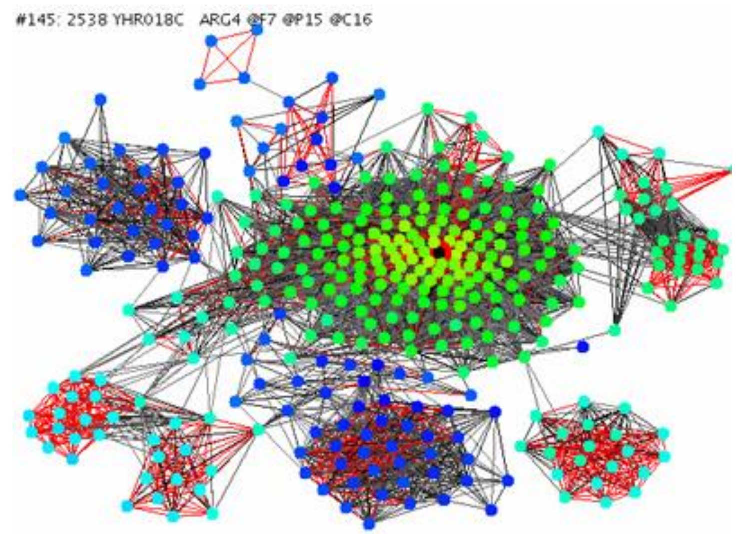
Distance (similarity) metric

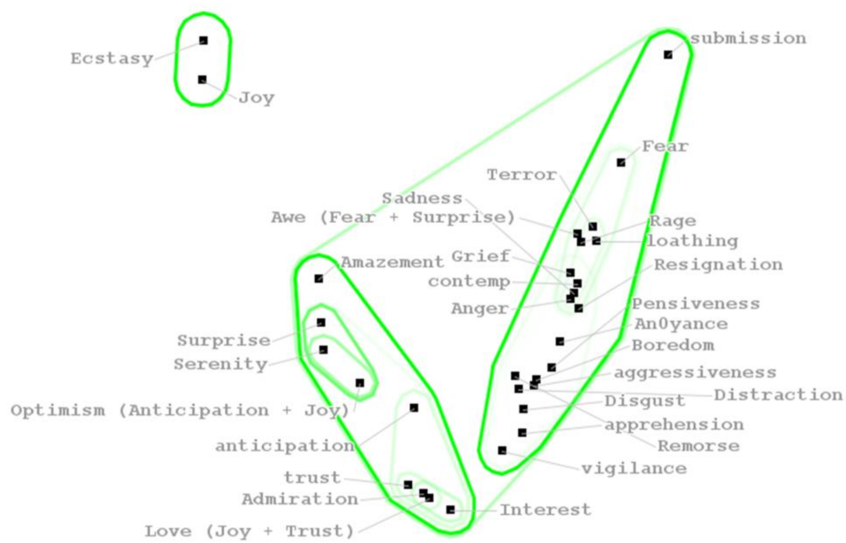
- Euclidian distance (best for data)
- Cosine distance (best for data)
- |1-correlation| distance (best for attributes)
- use 1-correlation to move correlated attribute points closer
- use | | if you do not care about positive or negative correlations

	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.

MDS Examples

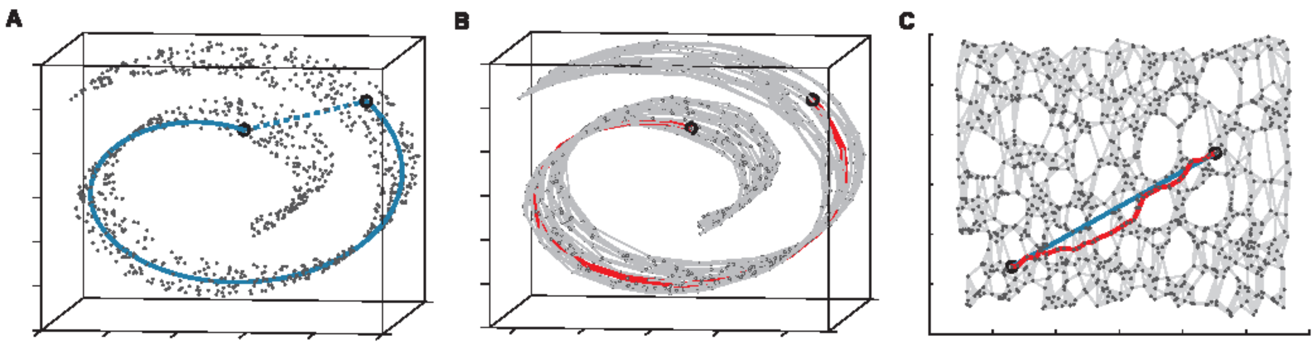






Manifold Learning: Isomap

by: [J. Tenenbaum, V. de Silva, J. Langford, Science, 2000]



Tries to unwrap a high-dimensional surface (A) → manifold

- noisy points could be averaged first and projected onto the manifold

Algorithm

- construct neighborhood graph G → (B)
- for each pair of points in G compute the shortest path distances → geodesic distances
- fill similarity matrix with these geodesic distances
- embed (layout) in low-D (2D) with MDS → (C)
- visualize it like an MDS layout

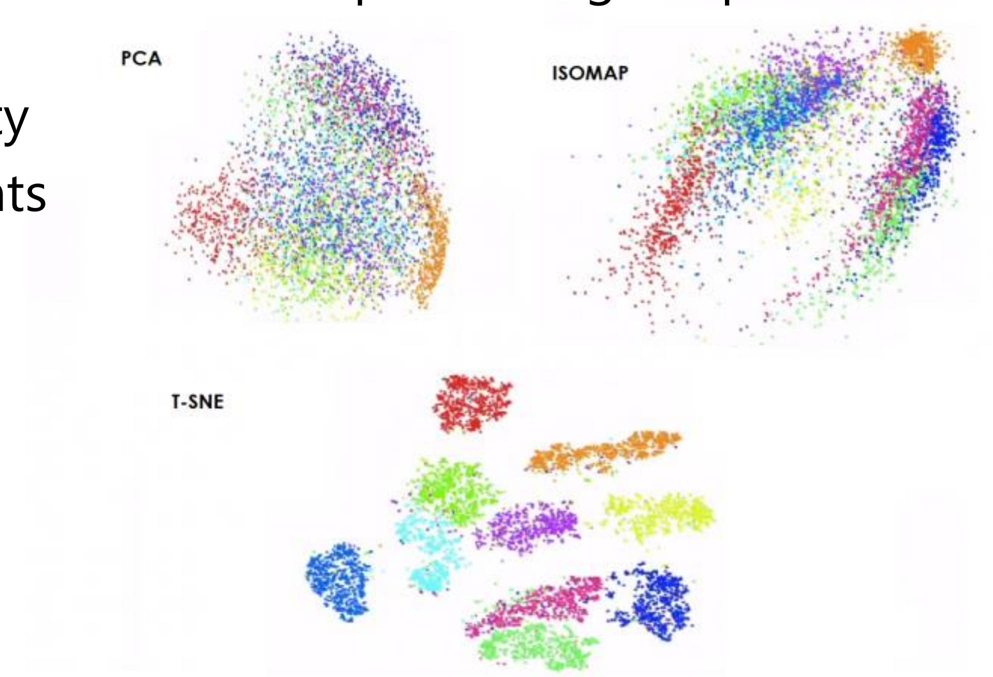
T-SNE

t-Distributed Stochastic Neighbor Embedding

- innovated by [l. van der Maaten and G. Hinton, 2008]

Works as a two-stage approach

1. Construct a probability distribution over pairs of high-D points based on similarity
2. Define a similar probability distribution over the points in the low-D map



Self-Organizing Maps (SOM)

Introduced by [T. Kohonen et al. 1996]

- unsupervised learning and clustering algorithm
- has advantages compared to hierarchical clustering
- often realized as an artificial neural network

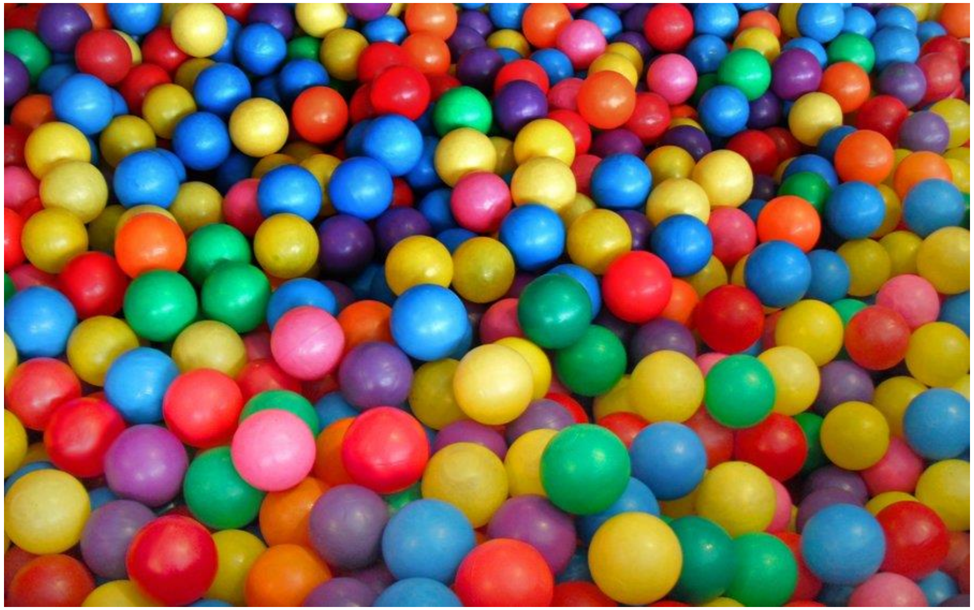
SOMs group the data

- perform a nonlinear projection from N-dimensional input space onto two-dimensional visualization space
- provide a useful topological arrangement of information objects in order to display clusters of similar objects in information space

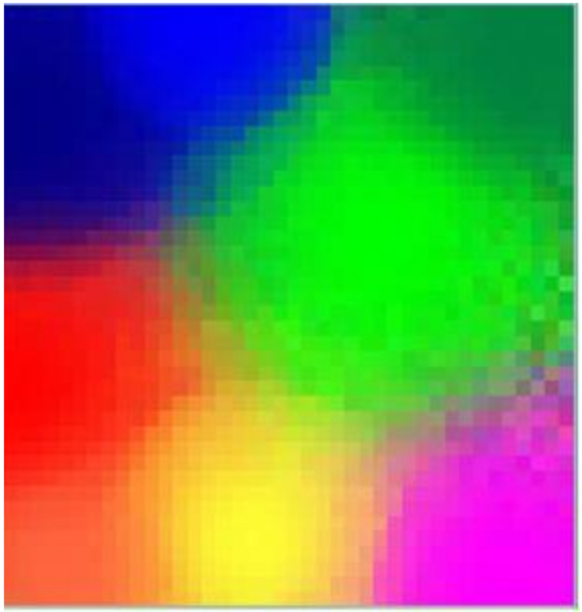
SOM Example

Map a dataset of 3D color vectors into a 2D plane

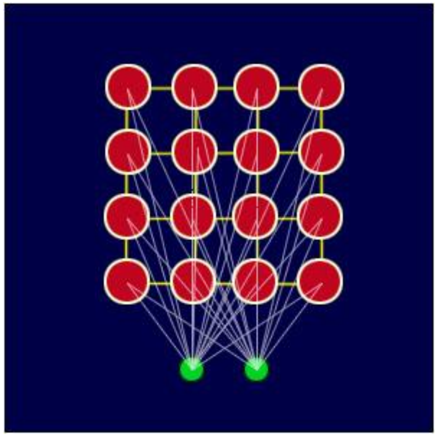
- assume you have an image with 5 colors
- want to see how many there are of each
- compute an SOM of the color vectors



SOM

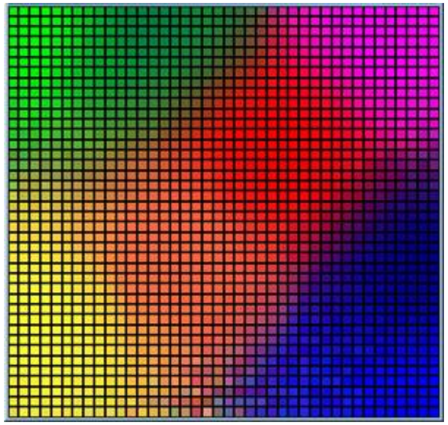


SOM Algorithm



Create array and connect all elements to the N input dimensions

- shown here: 2D vector with 4×4 elements
- initialize weights



For each input vector chosen at random

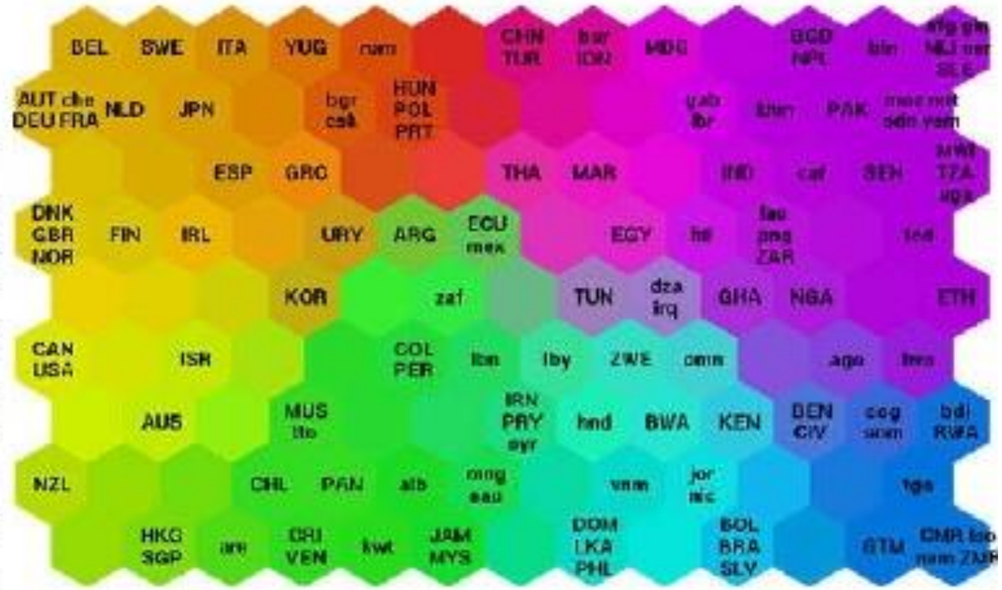
- find node with weights most like the input vector
- call that node the Best Matching Unit (BMU)
- find nodes within neighborhood radius *r* of BMU
    - initially *r* is chosen as the radius of the lattice
    - diminishes at each time step
- adjust the weights of the neighboring nodes to make them more like the input vector
    - the closer a node is to the BMU, the more its weights get altered

SOM Example: Poverty Map

SOM – Result Example

World Poverty Map

A SOM has been used to classify statistical data describing various quality-of-life factors such as state of health, nutrition, educational services etc. . **Countries with similar quality-of-life factors end up clustered together**. The countries with better quality-of-life are situated toward the upper left and the most poverty stricken countries are toward the lower right.

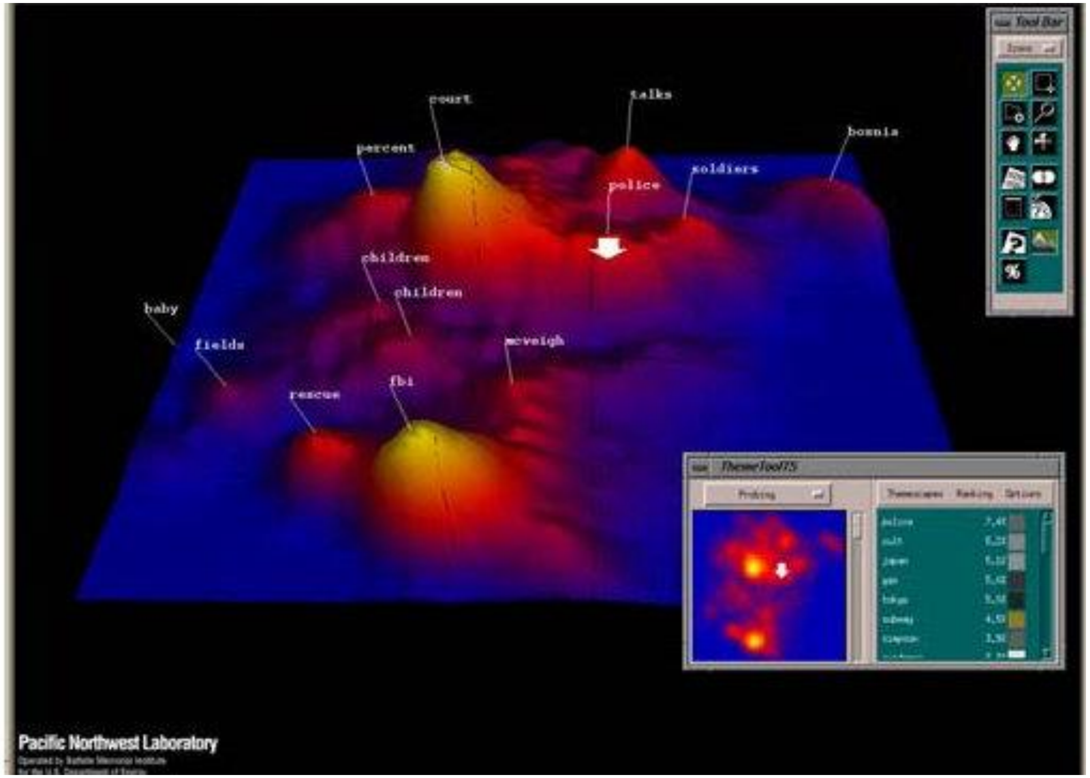


‘Poverty map’ based on 39 indicators from World Bank statistics (1992)

SOM Example: ThemeScape

Height represents density or number of documents in the region

Invented at Pacific Northwest National Lab (PNNL)



What About Categorical Variables?

You will need to use correspondence analysis (CA)

- CA is PCA for categorical variables
- related to factor analysis

Makes use of the χ^2 test

- what's χ^2 ?

Chi-square Test (Nominal Data)

A *chi-square test* is used to investigate relationships

Relationships between categorical, or nominal-scale, variables representing attributes of people, interaction techniques, systems, etc.

Data organized in a *contingency table* – cross tabulation containing counts (frequency data) for number of observations in each category

A chi-square test compares the *observed values* against *expected values*

Expected values assume “no difference”

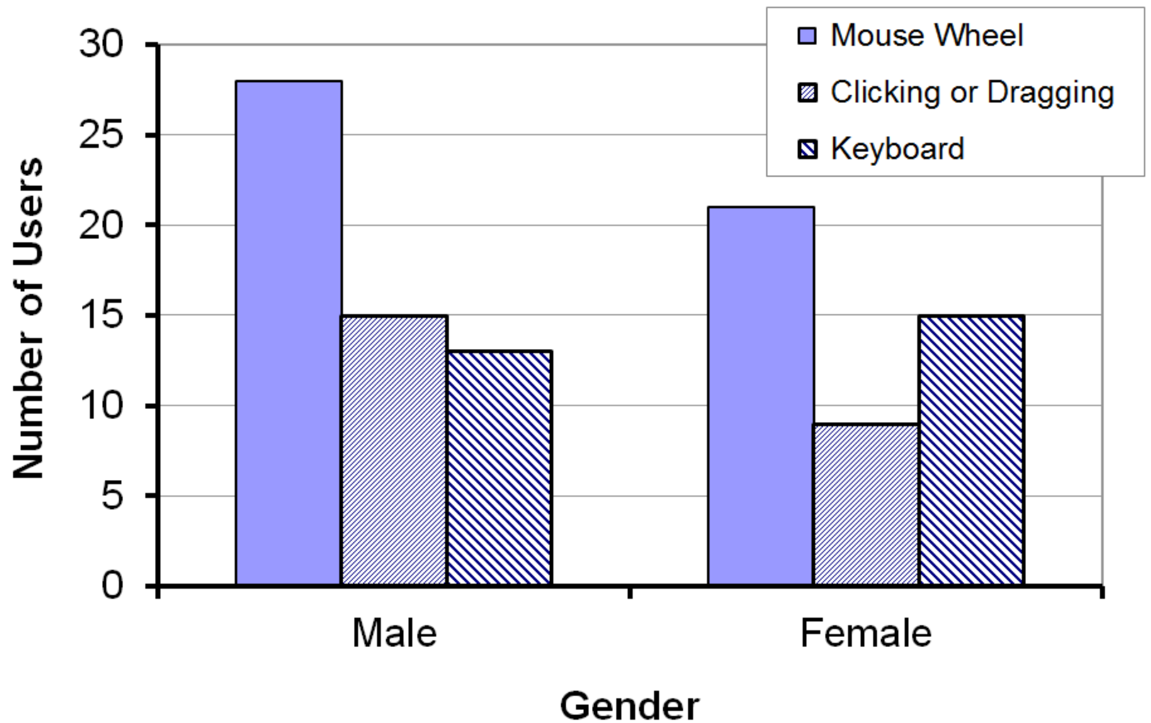
Research question:

- *Do males and females differ in their method of scrolling on desktop systems?* (next slide)

Chi-square – Example #1

Observed Number of Users				
Gender	Scrolling Method			Total
	MW	CD	KB	
Male	28	15	13	56
Female	21	9	15	45
Total	49	24	28	101

MW = mouse wheel
CD = clicking, dragging
KB = keyboard



Chi-square – Example #1

56.0·49.0/101=27.2

Expected Number of Users				
Gender	Scrolling Method			Total
	MW	CD	KB	
Male	27.2	13.3	15.5	56.0
Female	21.8	10.7	12.5	45.0
Total	49.0	24.0	28.0	101

(Observed-Expected)²/Expected = (28-27.2)²/27.2

Chi Squares				
Gender	Scrolling Method			Total
	MW	CD	KB	
Male	0.025	0.215	0.411	0.651
Female	0.032	0.268	0.511	0.811
Total	0.057	0.483	0.922	**1.462**

Significant if it exceeds critical value (next slide)

$$\chi^2 = 1.462$$

Chi-square Critical Values

Decide in advance on *alpha* (typically .05)

Degrees of freedom

- $df = (r - 1)(c - 1) = (2 - 1)(3 - 1) = 2$
- r = number of rows, c = number of columns

Significance Threshold (α)	Degrees of Freedom							
	1	2	3	4	5	6	7	8
.1	2.71	4.61	6.25	7.78	9.24	10.65	12.02	13.36
.05	3.84	$\boxed{5.99}$	7.82	9.49	11.07	12.59	14.07	15.51
.01	6.64	9.21	11.35	13.28	15.09	16.81	18.48	20.09
.001	10.83	13.82	16.27	18.47	20.52	22.46	24.32	26.13

$\chi^2 = 1.462$ (< 5.99 $\therefore$ not significant)

Correspondence Analysis (CA)

more info

Example:

	Smoking Category				
Staff Group	(1) None	(2) Light	(3) Medium	(4) Heavy	Row Totals
(1) Senior Managers	4	2	3	2	11
(2) Junior Managers	4	3	7	4	18
(3) Senior Employees	25	10	12	4	51
(4) Junior Employees	18	24	33	13	88
(5) Secretaries	10	6	7	2	25
Column Totals	61	45	62	25	193

There are two high-D spaces

- 4D (column) space spanned by smoking habits – plot staff group
- 5D (row) space spanned by staff group – plot smoking habits

Are these two spaces (the rows and columns) independent ?

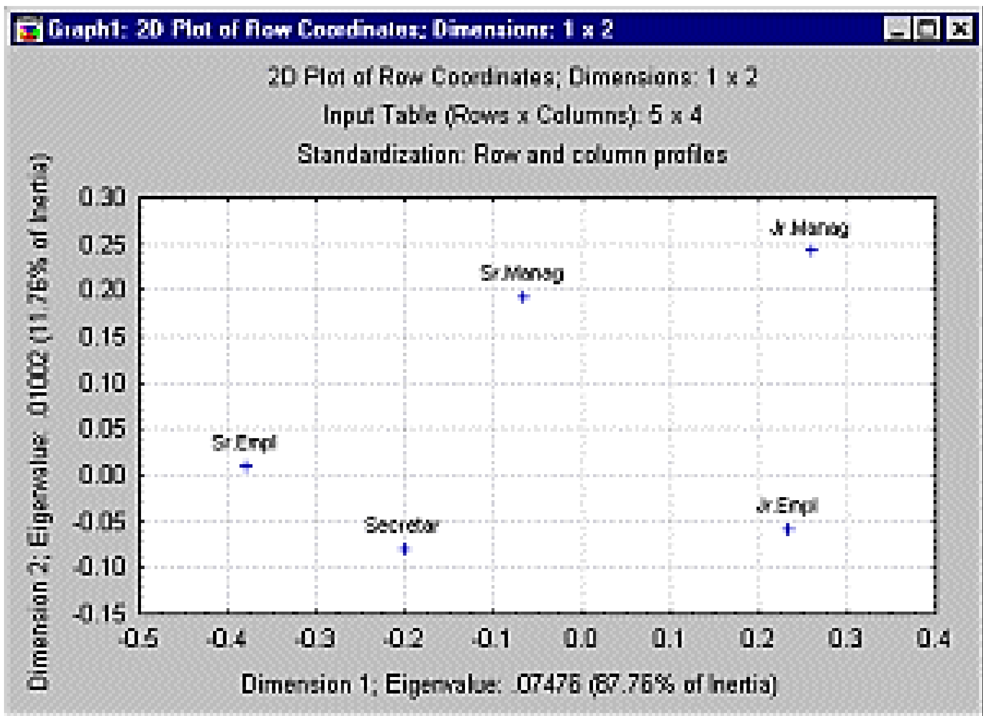
- this occurs when the χ^2 statistics of the table is insignificant

CA Eigen Analysis

	Smoking Category				
Staff Group	(1) None	(2) Light	(3) Medium	(4) Heavy	Row Totals
(1) Senior Managers	4	2	3	2	11
(2) Junior Managers	4	3	7	4	18
(3) Senior Employees	25	10	12	4	51
(4) Junior Employees	18	24	33	13	88
(5) Secretaries	10	6	7	2	25
Column Totals	61	45	62	25	193

Let's do some plotting

- compute distance matrix of the rows $\mathbf{CC}^T$
- compute Eigenvector matrix **U** and the Eigenvalue matrix **D**
- sort eigenvectors by values, pick two major vectors, create 2D plot



-- senior employees most similar to secretaries

Eigenvalues and Inertia for all Dimensions
Input Table (Rows x Columns): 5 x 4
Total Inertia = .08519 Chi² = 16.442

No. of Dims	Singular Values	Eigen-Values	Perc. of Inertia	Cumulatv Percent	Chi Squares
1	.273421	.074759	87.75587	87.7559	14.42851
2	.100086	.010017	11.75865	99.5145	1.93332
3	.020337	.000414	.48547	100.0000	.07982

CA Eigen Analysis

	Smoking Category				
Staff Group	(1) None	(2) Light	(3) Medium	(4) Heavy	Row Totals
(1) Senior Managers	4	2	3	2	11
(2) Junior Managers	4	3	7	4	18
(3) Senior Employees	25	10	12	4	51
(4) Junior Employees	18	24	33	13	88
(5) Secretaries	10	6	7	2	25
Column Totals	61	45	62	25	193

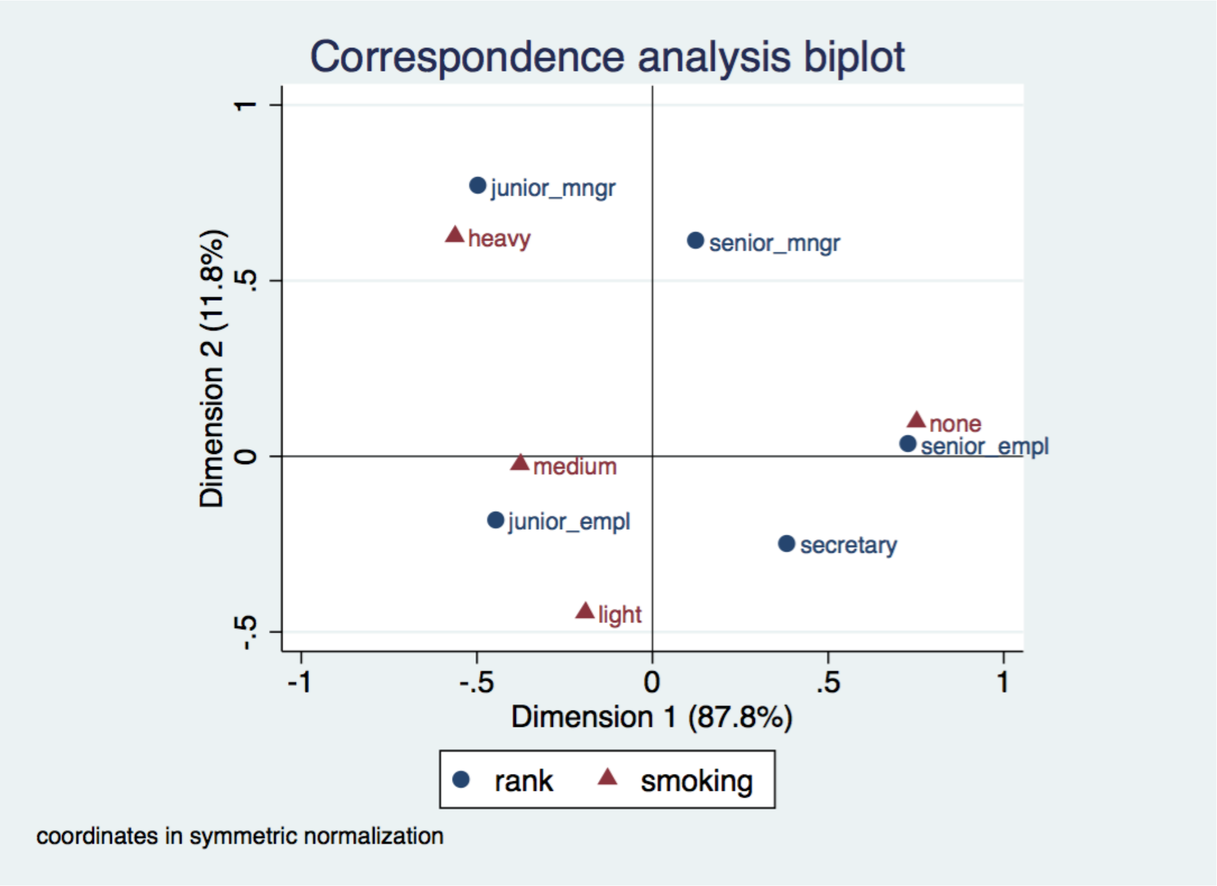
Next:

- compute distance matrix of the columns $\mathbf{C}^T\mathbf{C}$
- compute Eigenvector matrix **V** (gives the same Eigenvalue matrix **D**)
- sort eigenvectors by value
- pick two major vectors
- create 2D plot of smoking categories

Following (next slide):

- combine the plots of **U** and **V**
- if the χ^2 statistics was significant we should see some dependencies

Combined CA Plot



Interpretation sample (using the χ^2 frequentist mindset)

- *relatively speaking*, there are more non-smoking senior employees

Extending to Cases

Case Number	Senior Manager	Junior Manager	Senior Employee	Junior Employee	Secretary	None	Light	Medium	Heavy
1	1	0	0	0	0	1	0	0	0
2	1	0	0	0	0	1	0	0	0
3	1	0	0	0	0	1	0	0	0
4	1	0	0	0	0	1	0	0	0
5	1	0	0	0	0	0	1	0	0
...	.	.	.	.	.	.	.	.	.
...	.	.	.	.	.	.	.	.	.
...	.	.	.	.	.	.	.	.	.
191	0	0	0	0	1	0	0	1	0
192	0	0	0	0	1	0	0	0	1
193	0	0	0	0	1	0	0	0	1

Plot would now show 193 cases and 9 variables

Multiple Correspondence Analysis

Extension where there are more than 2 categorical variables

	SURVIVAL		AGE			LOCATION		
Case No.	NO	YES	LESST50	A50TO69	OVER69	TOKYO	BOSTON	GLAMORGN
1	0	1	0	1	0	0	0	1
2	1	0	1	0	0	1	0	0
3	0	1	0	1	0	0	1	0
4	0	1	0	0	1	0	0	1
...	.	.	.	.	.	.	.	.
...	.	.	.	.	.	.	.	.
...	.	.	.	.	.	.	.	.
762	1	0	0	1	0	1	0	0
763	0	1	1	0	0	0	1	0
764	0	1	0	1	0	0	0	1

Let’s call it matrix X

Multiple Correspondence Analysis

Compute X'X to get the Burt Table

	SURVIVAL		AGE			LOCATION		
	NO	YES	<50	50-69	69+	TOKYO	BOSTON	GLAMORGN
SURVIVAL:NO	210	0	68	93	49	60	82	68
SURVIVAL:YES	0	554	212	258	84	230	171	153
AGE:UNDER_50	68	212	280	0	0	151	58	71
AGE:A_50TO69	93	258	0	351	0	120	122	109
AGE:OVER_69	49	84	0	0	133	19	73	41
LOCATION:TOKYO	60	230	151	120	19	290	0	0
LOCATION:BOSTON	82	171	58	122	73	0	253	0
LOCATION:GLAMORGN	68	153	71	109	41	0	0	221

Compute Eigenvectors and Eigenvalues

- keep top two Eigenvectors/values
- visualize the attribute loadings of these two Eigenvectors into the Burt table plot (the loadings are the coordinates)

Larger MCA Example

Results of a survey of car owners and car attributes

Burt Table

	American	European	Japanese	Large	Medium	Small	Family	Sporty	Work	1 Income	2 Incomes	Own	Rent	Married	Married with Kids	Single	Single with Kids	Female	Male
American	125	0	0	36	60	29	81	24	20	58	67	93	32	37	50	32	6	58	67
European	0	44	0	4	20	20	17	23	4	18	26	38	6	13	15	15	1	21	23
Japanese	0	0	165	2	61	102	76	59	30	74	91	111	54	51	44	62	8	70	95
Large	36	4	2	42	0	0	30	1	11	20	22	35	7	9	21	11	1	17	25
Medium	60	20	61	0	141	0	89	39	13	57	84	106	35	42	51	40	8	70	71
Small	29	20	102	0	0	151	55	66	30	73	78	101	50	50	37	58	6	62	89
Family	81	17	76	30	89	55	174	0	0	69	105	130	44	50	79	35	10	83	91
Sporty	24	23	59	1	39	66	0	106	0	55	51	71	35	35	12	57	2	44	62
Work	20	4	30	11	13	30	0	0	54	26	28	41	13	16	18	17	3	22	32
1 Income	58	18	74	20	57	73	69	55	26	150	0	80	70	10	27	99	14	47	103
2 Incomes	67	26	91	22	84	78	105	51	28	0	184	162	22	91	82	10	1	102	82
Own	93	38	111	35	106	101	130	71	41	80	162	242	0	76	106	52	8	114	128
Rent	32	6	54	7	35	50	44	35	13	70	22	0	92	25	3	57	7	35	57
Married	37	13	51	9	42	50	50	35	16	10	91	76	25	101	0	0	0	53	48
Married with Kids	50	15	44	21	51	37	79	12	18	27	82	106	3	0	109	0	0	48	61
Single	32	15	62	11	40	58	35	57	17	99	10	52	57	0	0	109	0	35	74
Single with Kids	6	1	8	1	8	6	10	2	3	14	1	8	7	0	0	0	15	13	2
Female	58	21	70	17	70	62	83	44	22	47	102	114	35	53	48	35	13	149	0
Male	67	23	95	25	71	89	91	62	32	103	82	128	57	48	61	74	2	0	185

more info see [here]

MCA Example (2)

Summary table:

Inertia and Chi-Square Decomposition

Singular Value	Principal Inertia	Chi-Square	Percent	Cumulative Percent	4 8 12 16 20 ----+----+----+----+----+---
0.56934	0.32415	970.77	18.91	18.91	************************
0.48352	0.23380	700.17	13.64	32.55	*****************
0.42716	0.18247	546.45	10.64	43.19	*************
0.41215	0.16987	508.73	9.91	53.10	************
0.38773	0.15033	450.22	8.77	61.87	***********
0.38520	0.14838	444.35	8.66	70.52	***********
0.34066	0.11605	347.55	6.77	77.29	********
0.32983	0.10879	325.79	6.35	83.64	********
0.31517	0.09933	297.47	5.79	89.43	*******
0.28069	0.07879	235.95	4.60	94.03	******
0.26115	0.06820	204.24	3.98	98.01	*****
0.18477	0.03414	102.24	1.99	100.00	**
Total	1.71429	5133.92	100.00		

Degrees of Freedom = 324

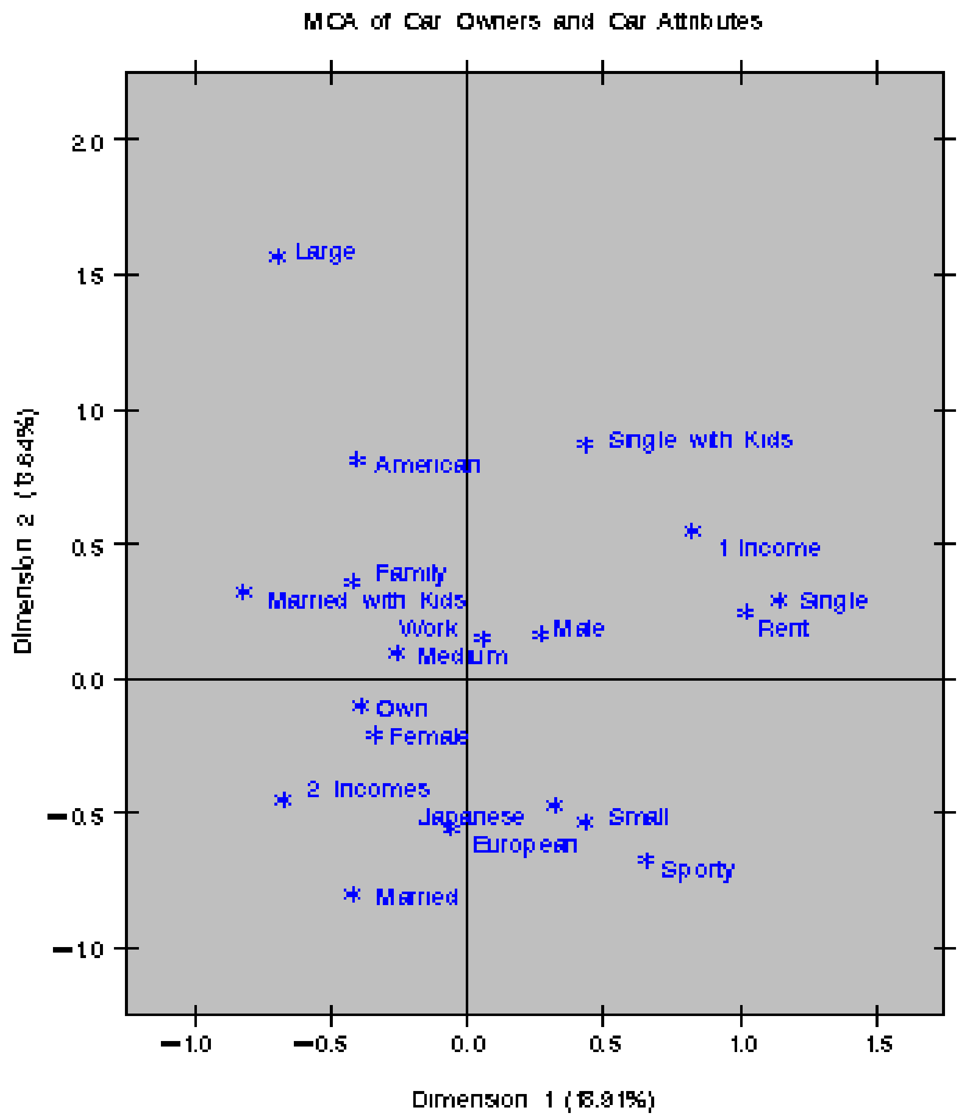
MCA Example (3)

Most influential column points (loadings):

Column Coordinates	Dim1	Dim2
American	-0.4035	0.8129
European	-0.0568	-0.5552
Japanese	0.3208	-0.4678
Large	-0.6949	1.5666
Medium	-0.2562	0.0965
Small	0.4326	-0.5258
Family	-0.4201	0.3602
Sporty	0.6604	-0.6696
Work	0.0575	0.1539
1 Income	0.8251	0.5472
2 Incomes	-0.6727	-0.4461
Own	-0.3887	-0.0943
Rent	1.0225	0.2480
Married	-0.4169	-0.7954
Married with Kids	-0.8200	0.3237
Single	1.1461	0.2930
Single with Kids	0.4373	0.8736
Female	-0.3365	-0.2057
Male	0.2710	0.1656

MCA Example (4)

Burt table plot:



Plot Observations

Top-right quadrant:

- categories single, single with kids, 1 income, and renting a home are associated

Proceeding clockwise:

- the categories sporty, small, and Japanese are associated
- being married, owning your own home, and having two incomes are associated
- having children is associated with owning a large American family car

Such information could be used in market research to identify target audiences for advertisements

Gartner Magic Quadrant

A Gartner Magic Quadrant is a culmination of research in a specific market, providing a wide-angle view of the relative positions of the market's competitors

This concept can be used for other dimension pairs as well

- essentially require to think of a segmentation of the 4 quadrants

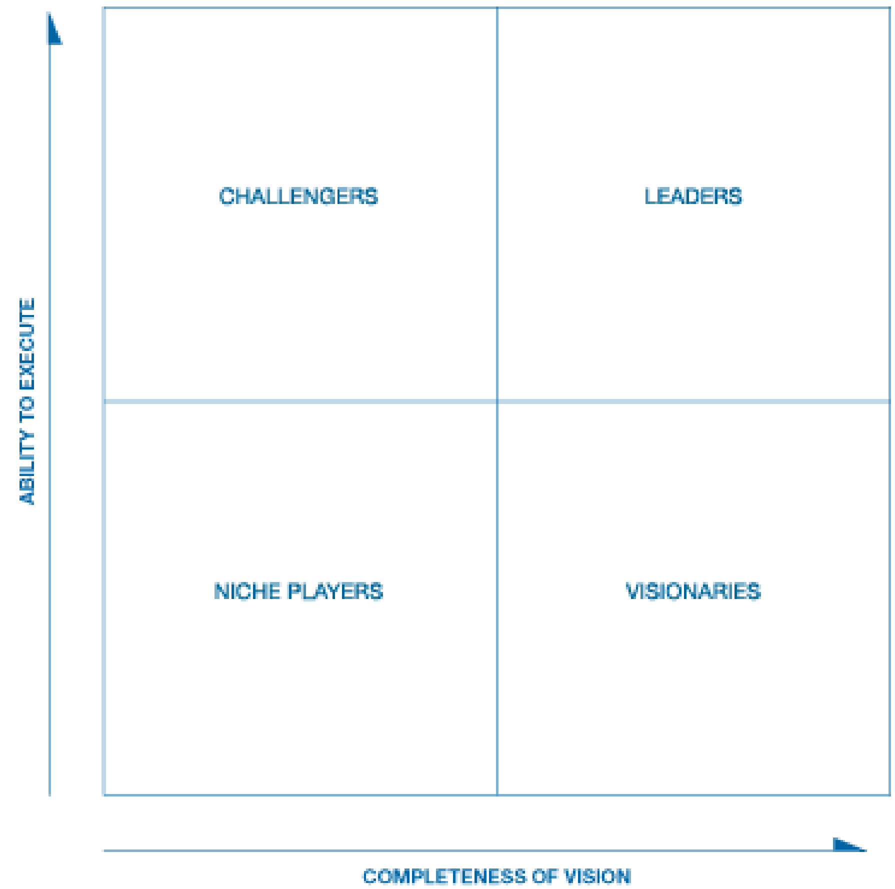
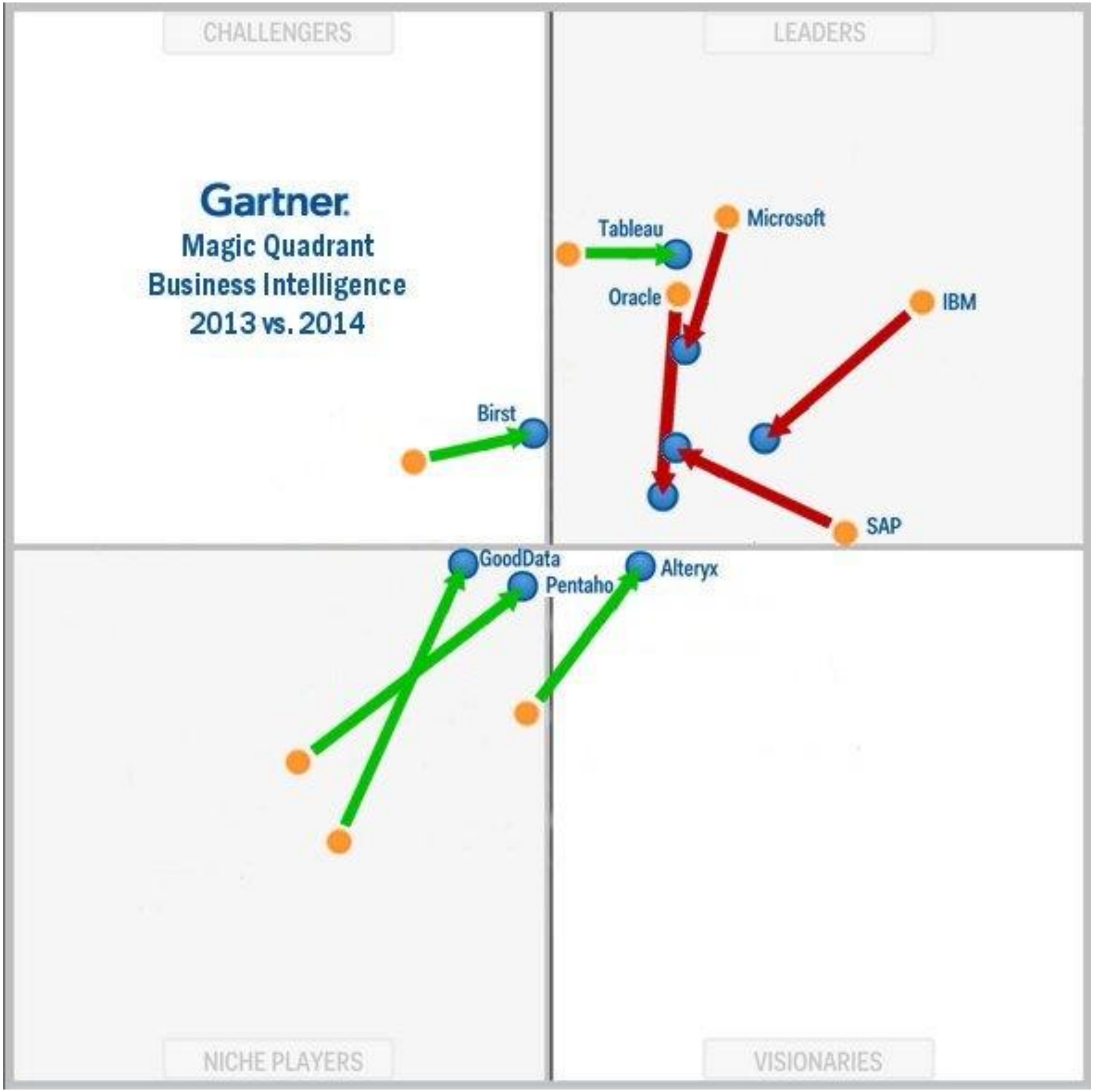


Figure 1. Magic Quadrant for Business Intelligence and Analytics Platforms

Source: Gartner (February 2014)



CHALLENGERS
LEADERS
Gartner
Magic Quadrant
Business Intelligence
2013 vs. 2014
Tableau
Microsoft
Oracle
IBM
Birst
SAP
GoodData
Pentaho
Alteryx
NICHE PLAYERS
VISIONARIES

Notes on Project #2

Submission site is not operational at this point

- turns out Blackboard supports peer review as well
- will use Blackboard for **report and video only**
- will use Google forms submission for source code

Video recording

- a good program is Apowersoft Screen Recorder
- captures screen and voice at the same time
- it's free for a version with sufficient capabilities