

# MULTIDIMENSIONAL DATA II

# RECOLLECTION

## SUBSETTING

Scatterplots, hyperslice,...

## EMBEDDING

Color, glyphs, worlds-within-worlds,...

## REDUCTION

SOM, MDS, PCA, RadViz,...

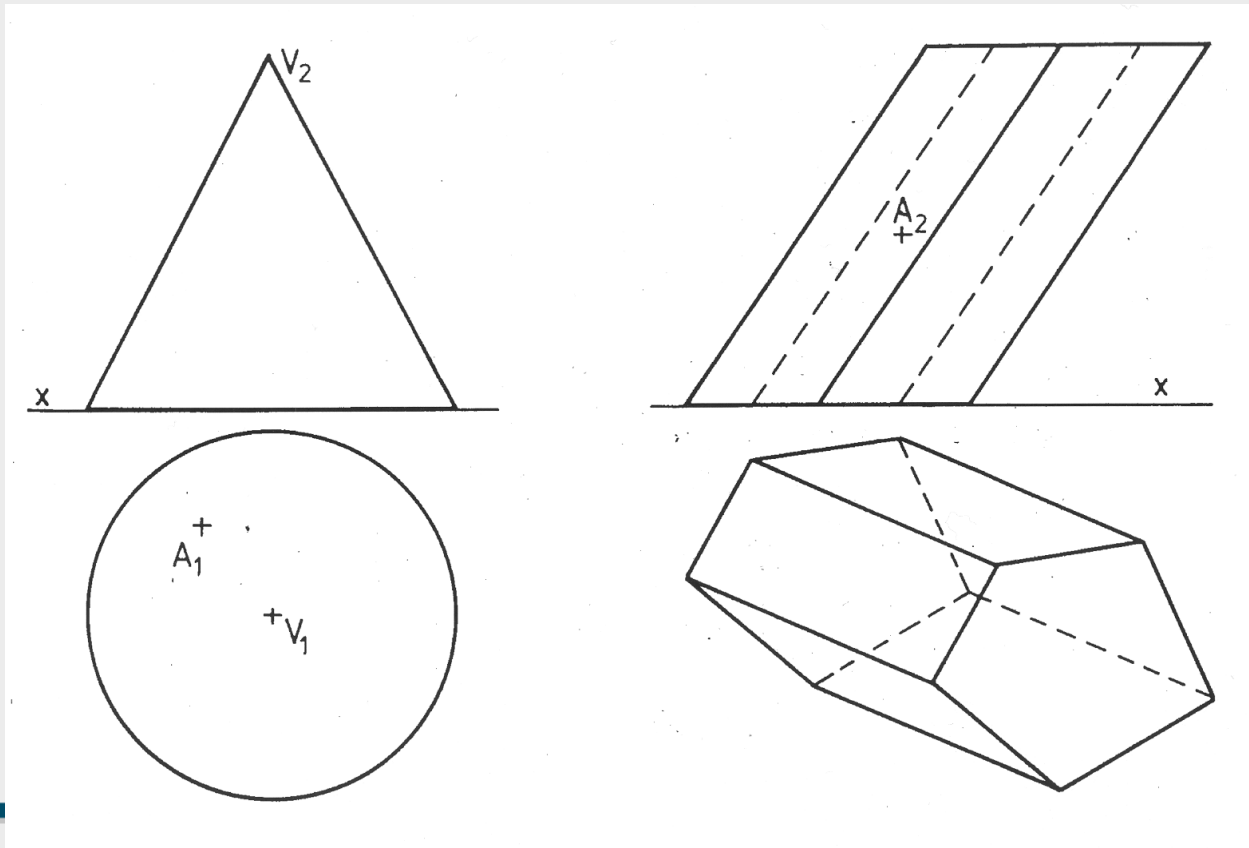
## AXIS RECONFIGURATION

---

# AXIS RECONFIGURATION

LEAVING CARTESIAN COORDINATES

EXAMPLE: MONGE PROJECTION



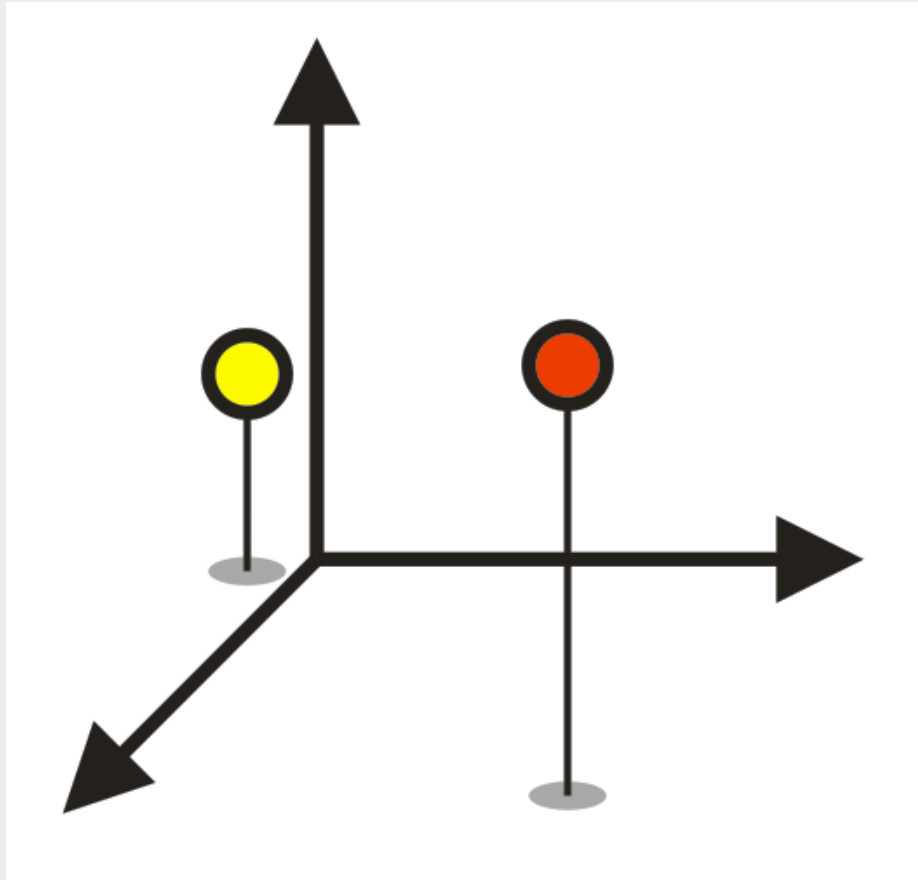
# MOTIVATION

## USUAL $ND \rightarrow 2D$ PROJECTION

Works only for small  $n$

Ambiguous

Point  $\leftrightarrow$  point



# SIMPLE RECONFIGURATION

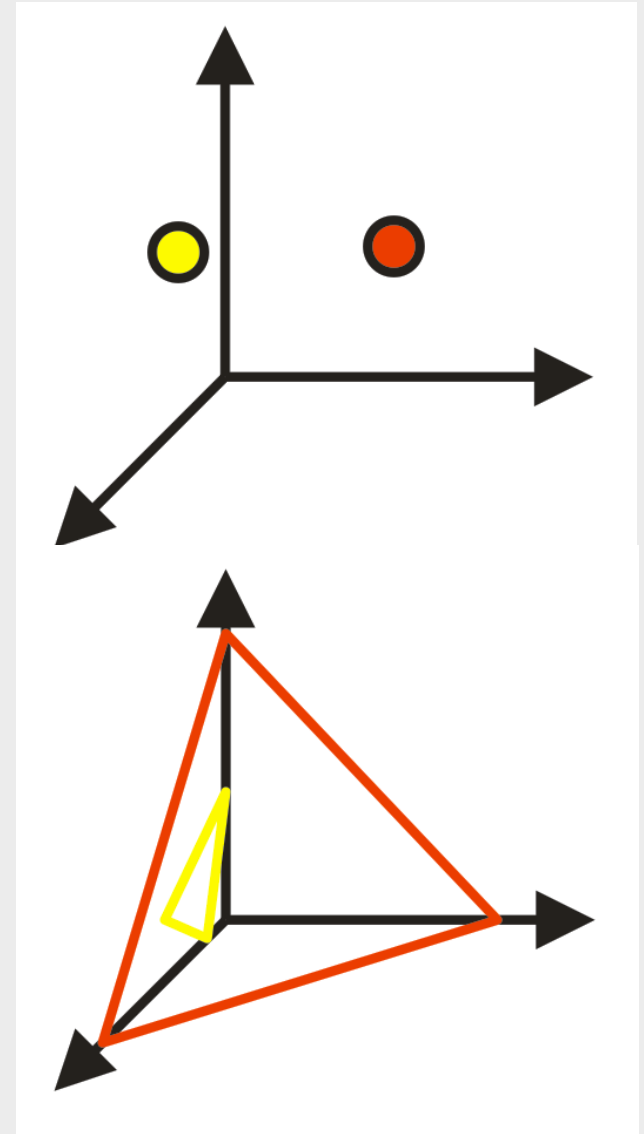
POINT  $\leftrightarrow$  LINE

SCALABLE TO LARGE  $N$

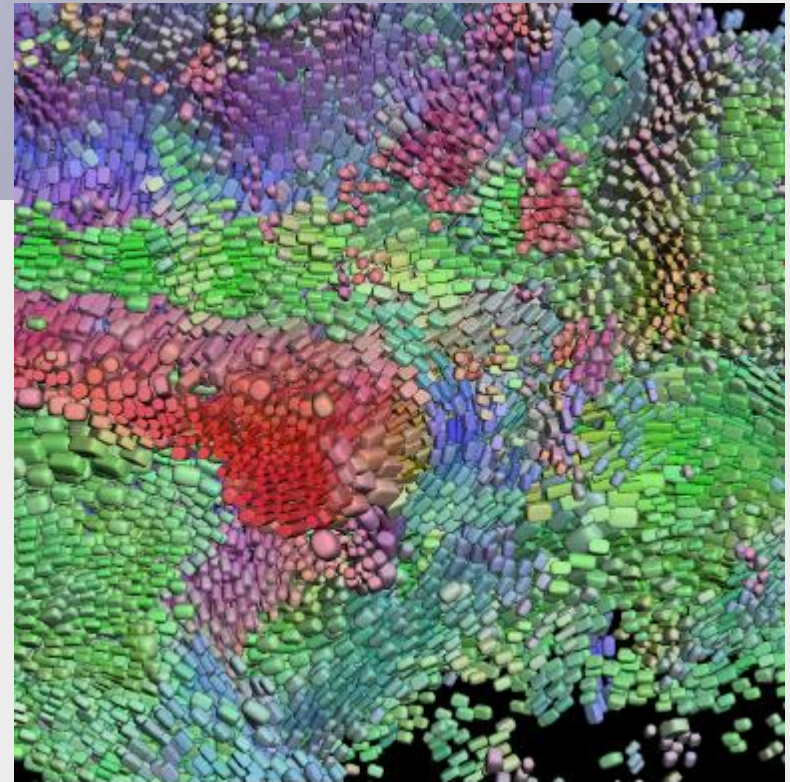
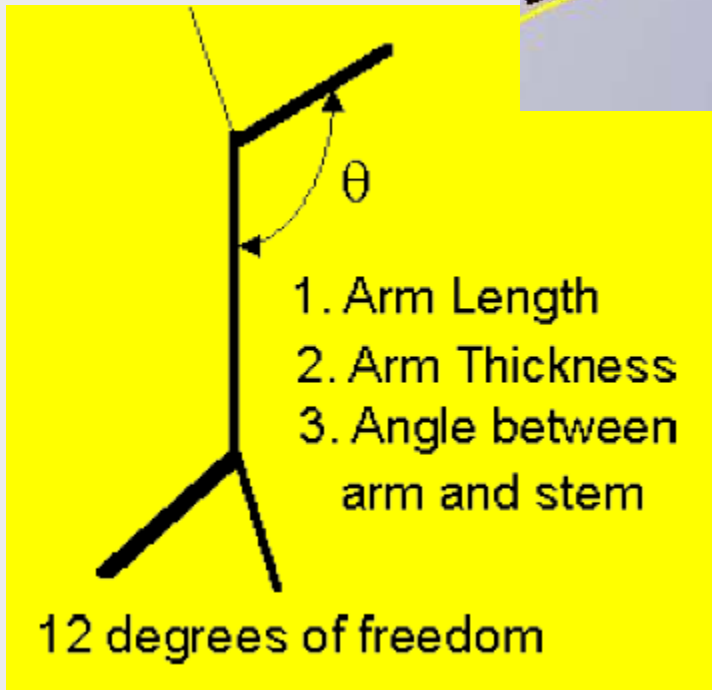
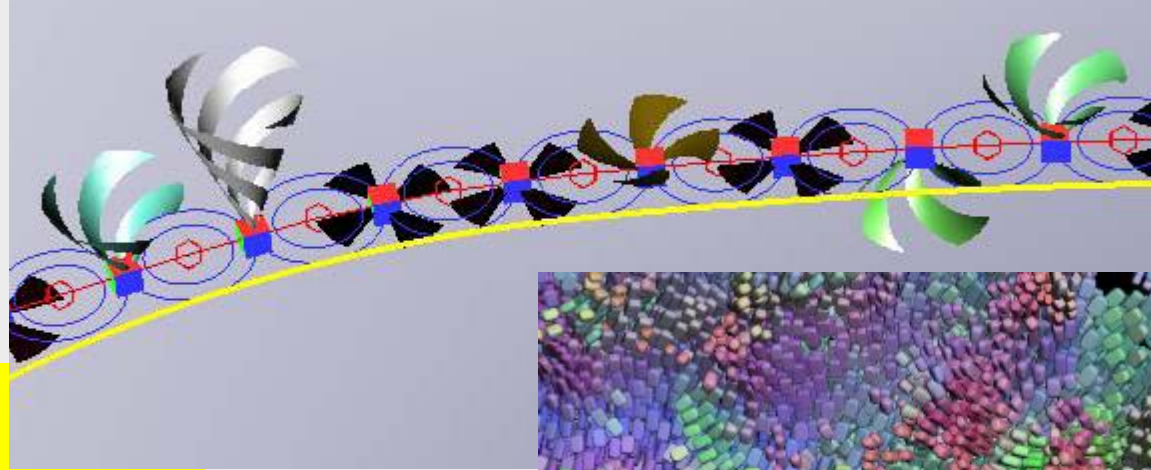
UNAMBIGUOUS

PRESERVES SIMILARITIES

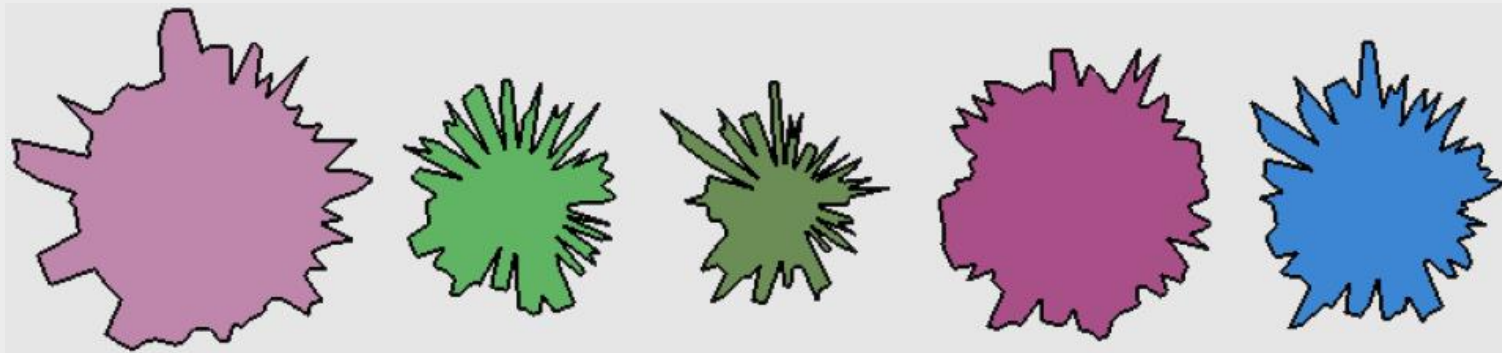
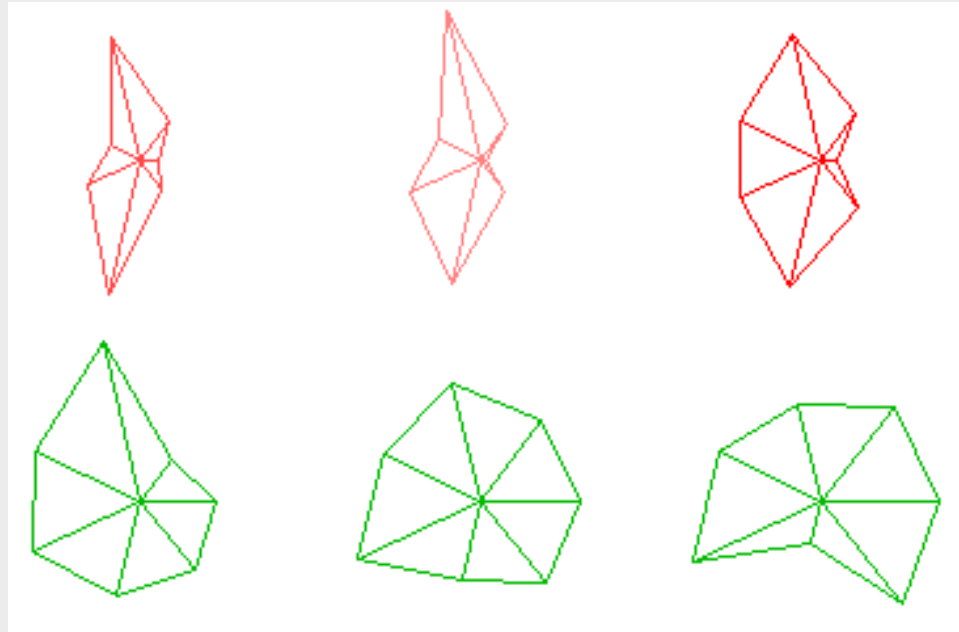
HARDER TO READ



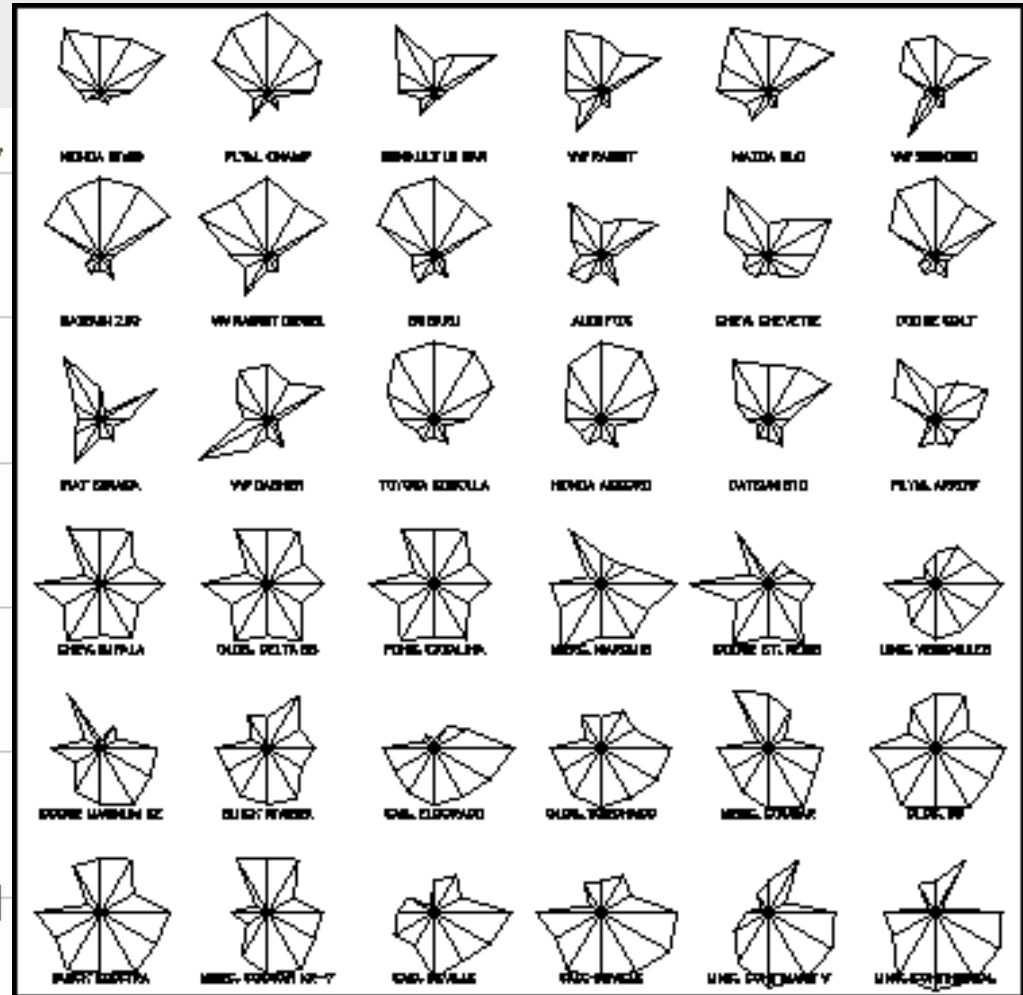
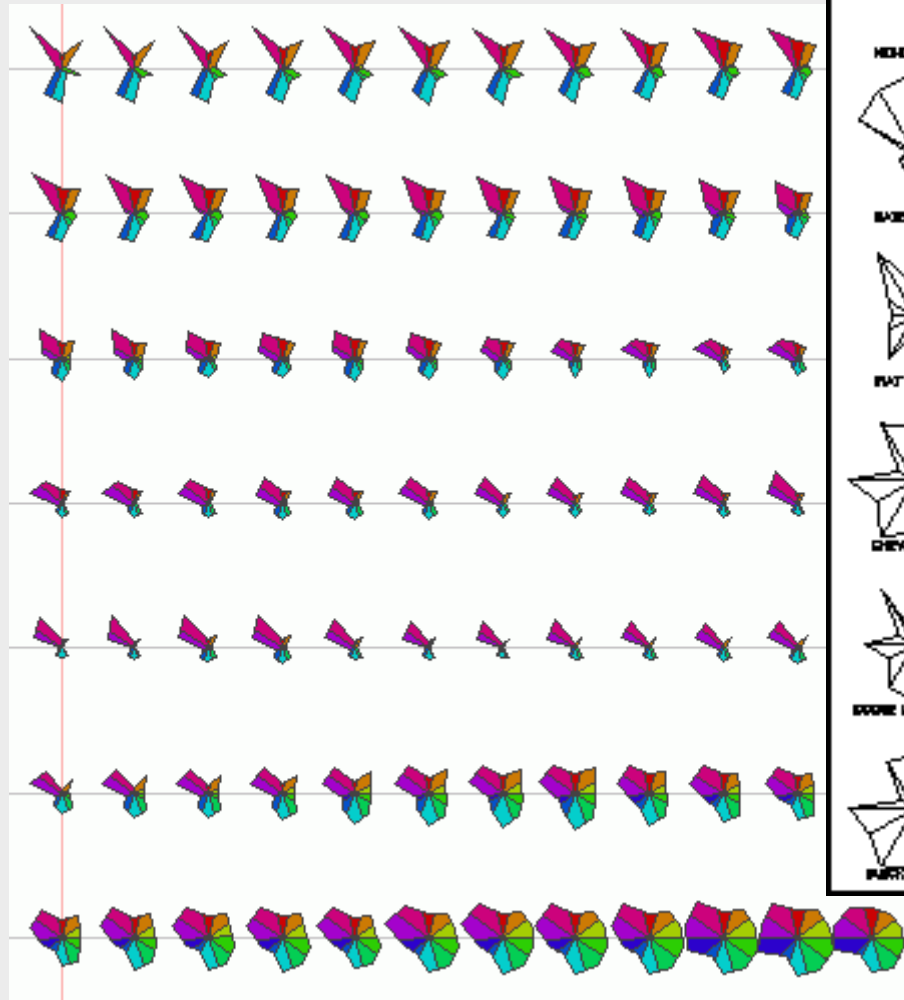
# RECOLLECTION: GLYPHS



# STAR GLYPH / STAR PLOT

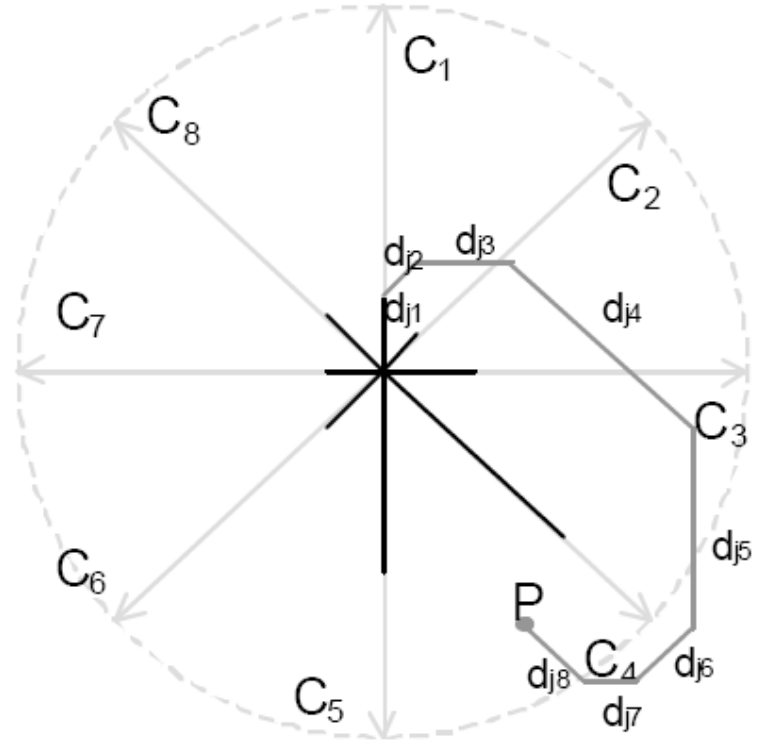
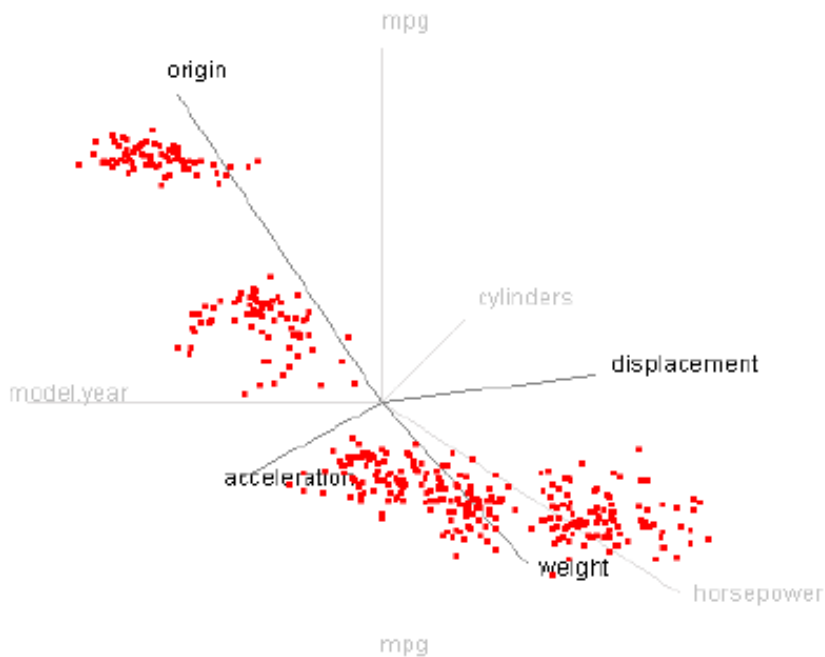


# STAR PLOT EXAMPLES



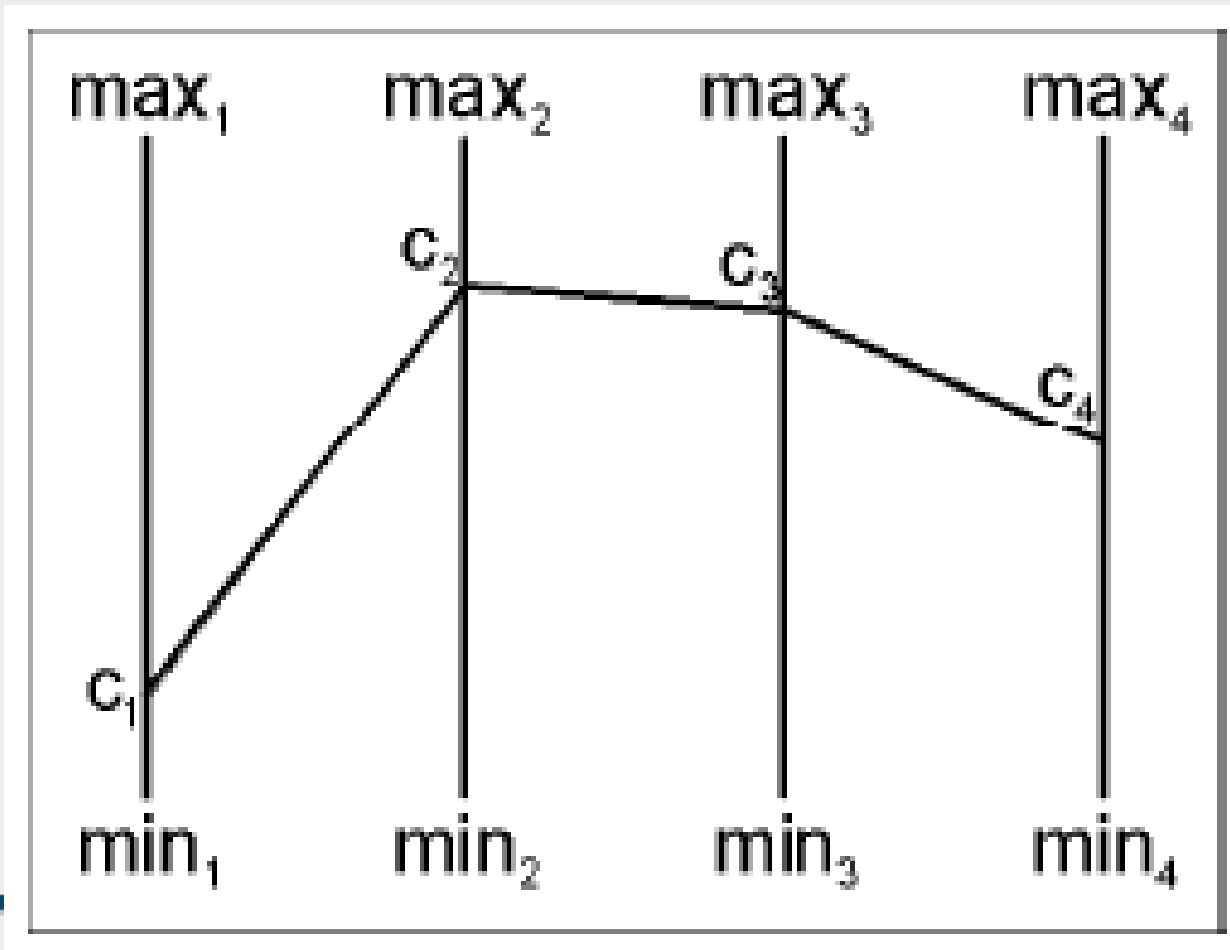


# STAR COORDINATES



# PARALLEL COORDINATES

ALFRED INSELBERG  
POINT  $\leftrightarrow$  LINE DUALITY



# POINT-LINE DUALITY IN PAR.COORDS.

PARALLEL

LINE  $p$

POINT  $M$

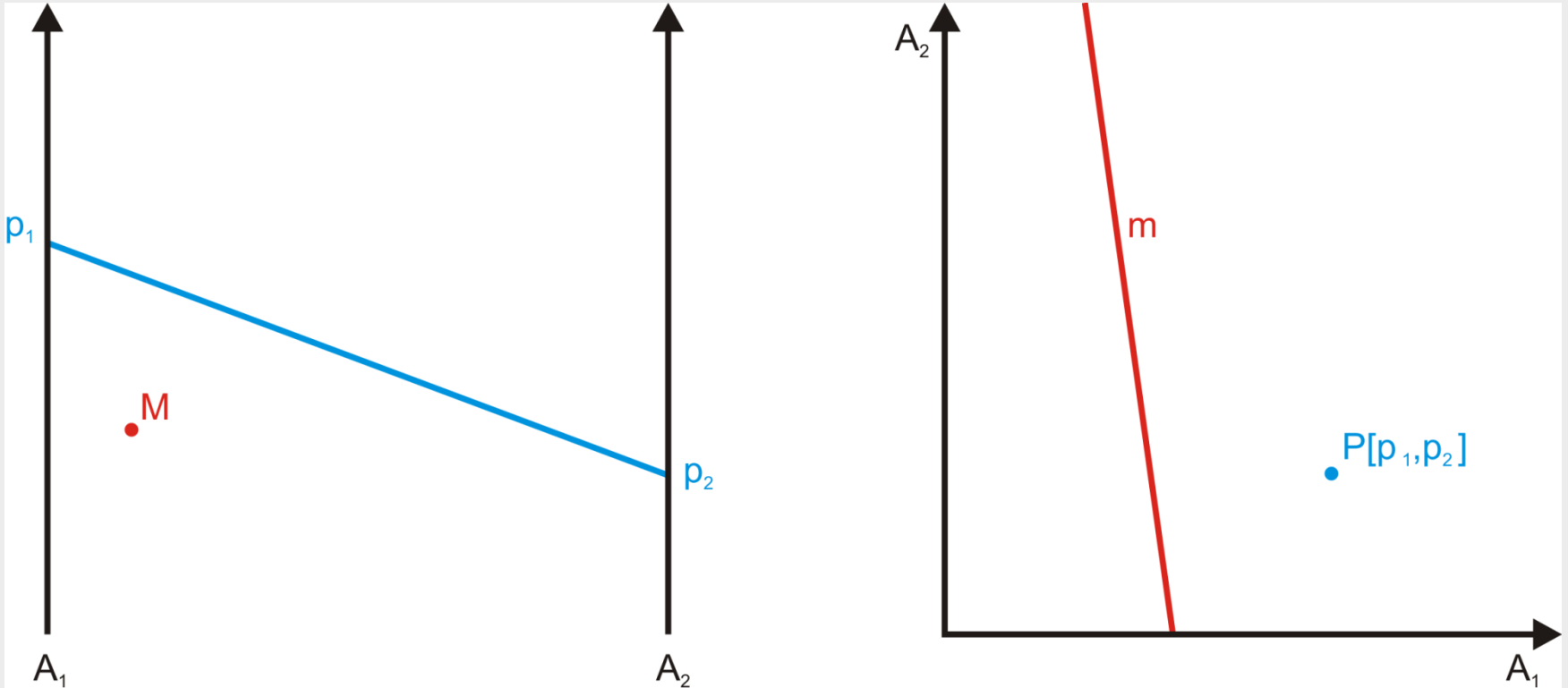
...

...

CARTESIAN

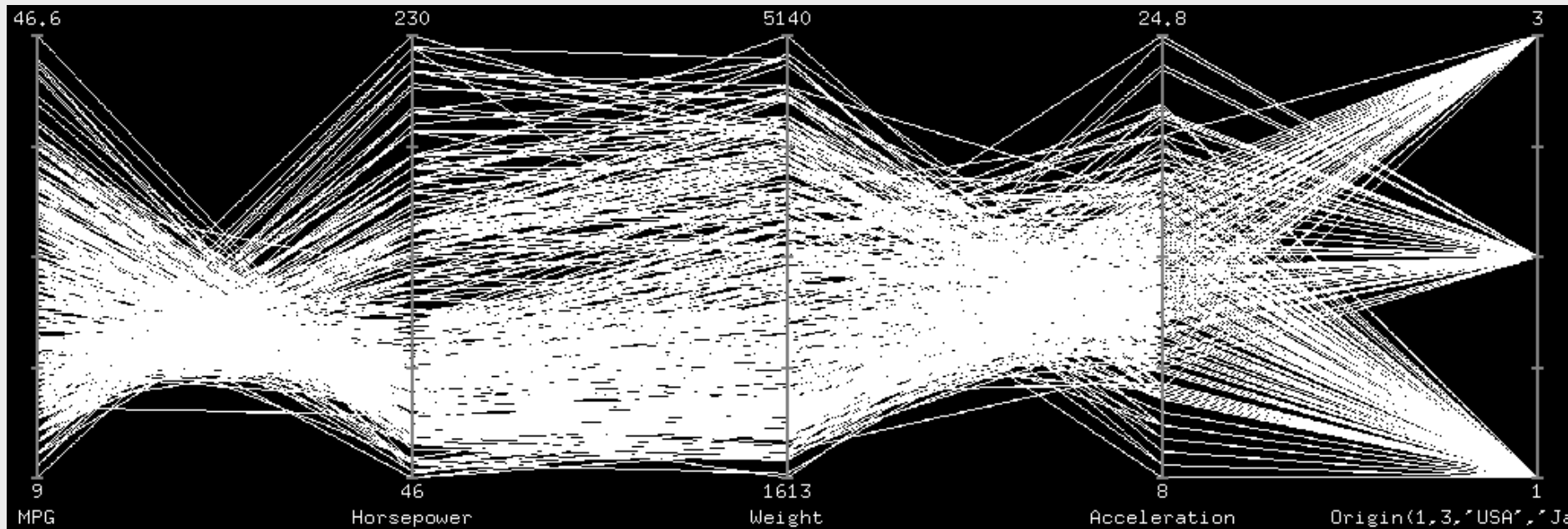
POINT  $P$

LINE  $m$



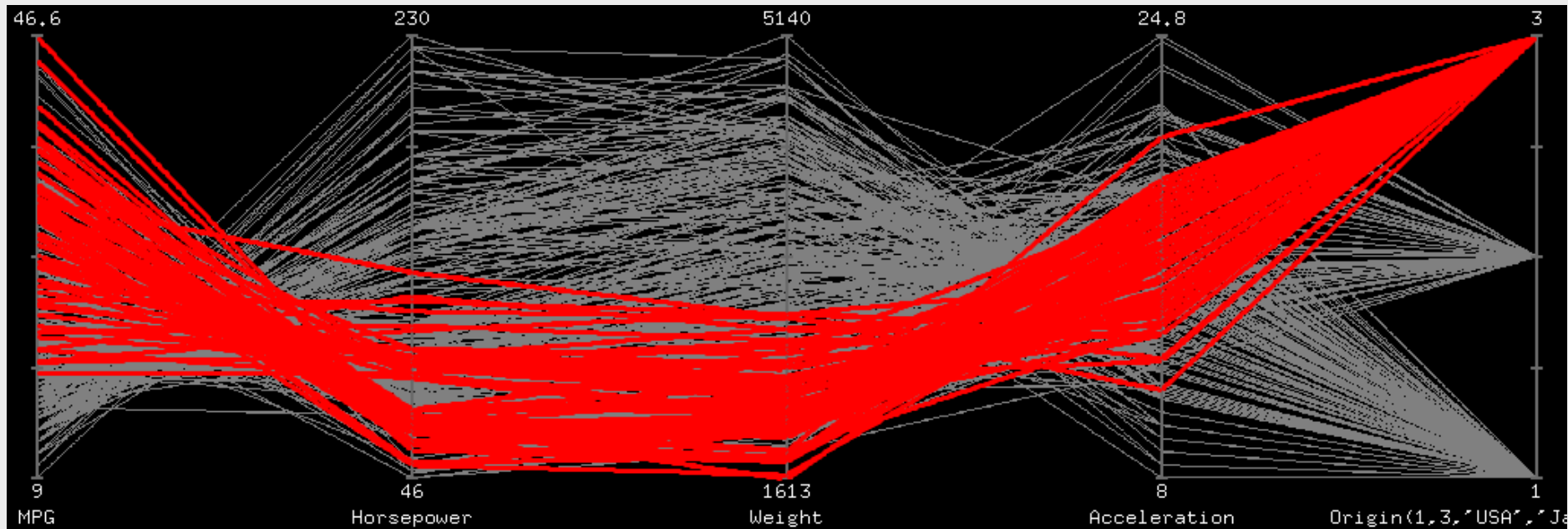
# VISUAL ANALYSIS IN PAR.COORDS.

## CORRELATIONS, PATTERNS



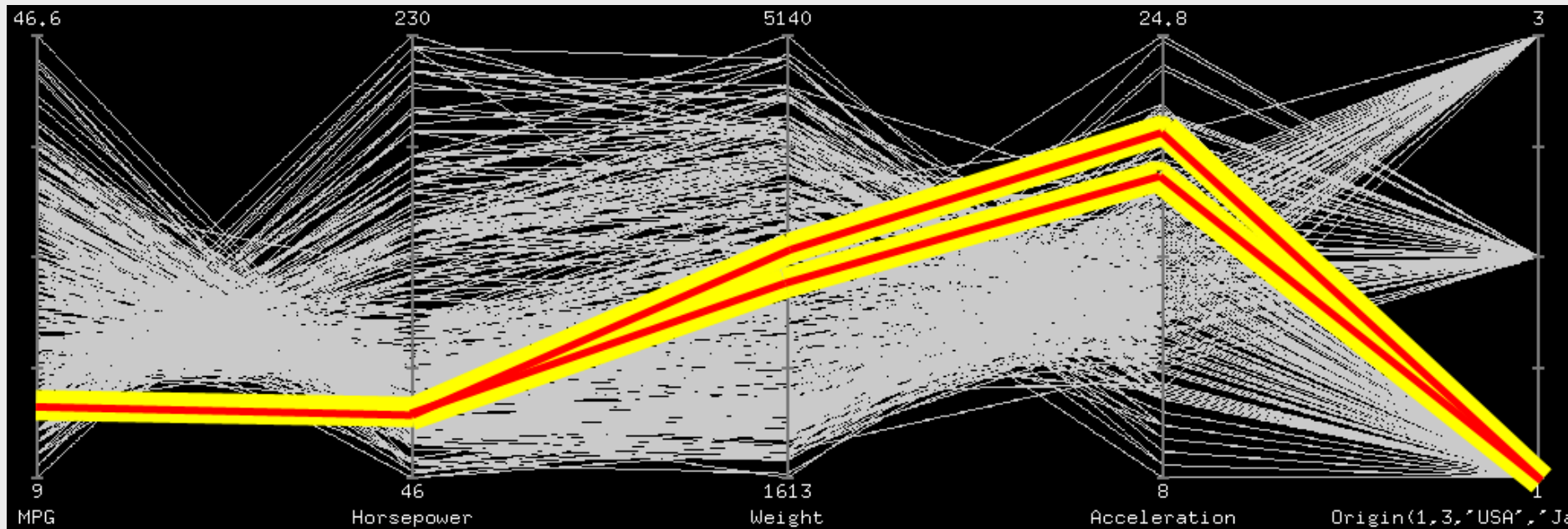
# VISUAL ANALYSIS IN PAR.COORDS.

## GROUPS



# VISUAL ANALYSIS IN PAR.COORDS.

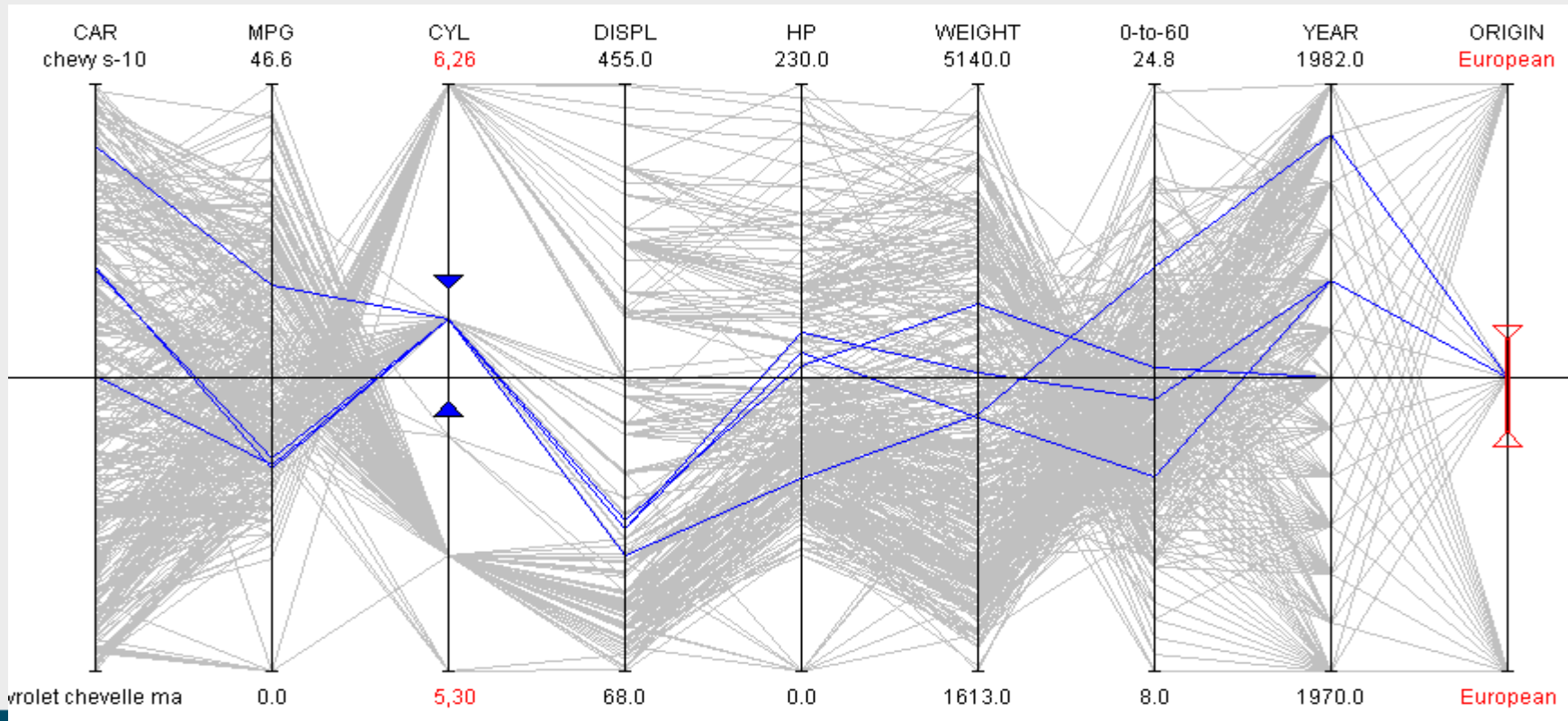
## OUTLIERS



# INTERACTION WITH PAR.COORDS.

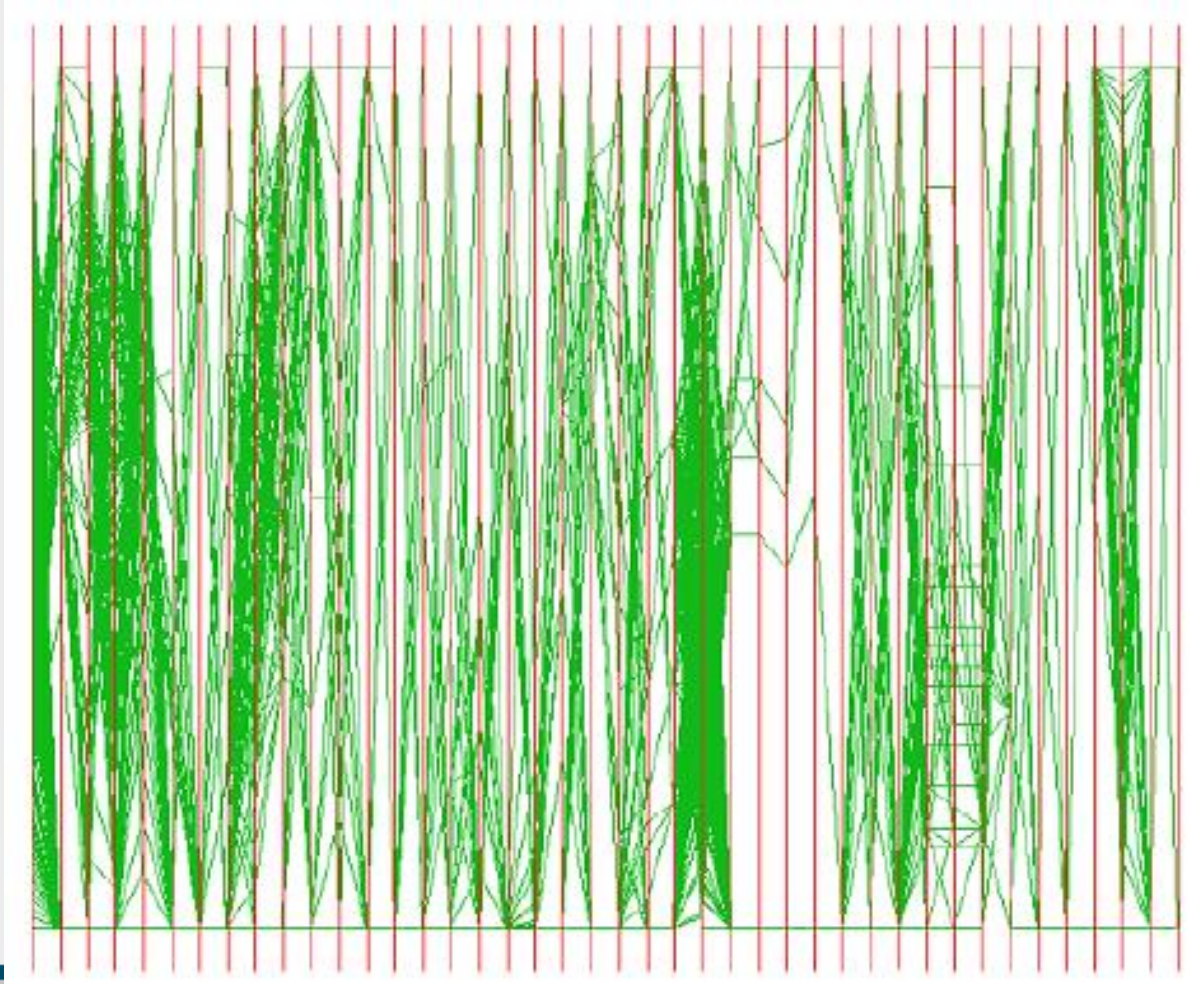
## AXIS ORDER, AXIS ORIENTATION BRUSHING

Combinations of per axis brushes



# PARALLEL COORDINATES PROPERTIES

SCALES UP TO  $N = 10 \dots 20$



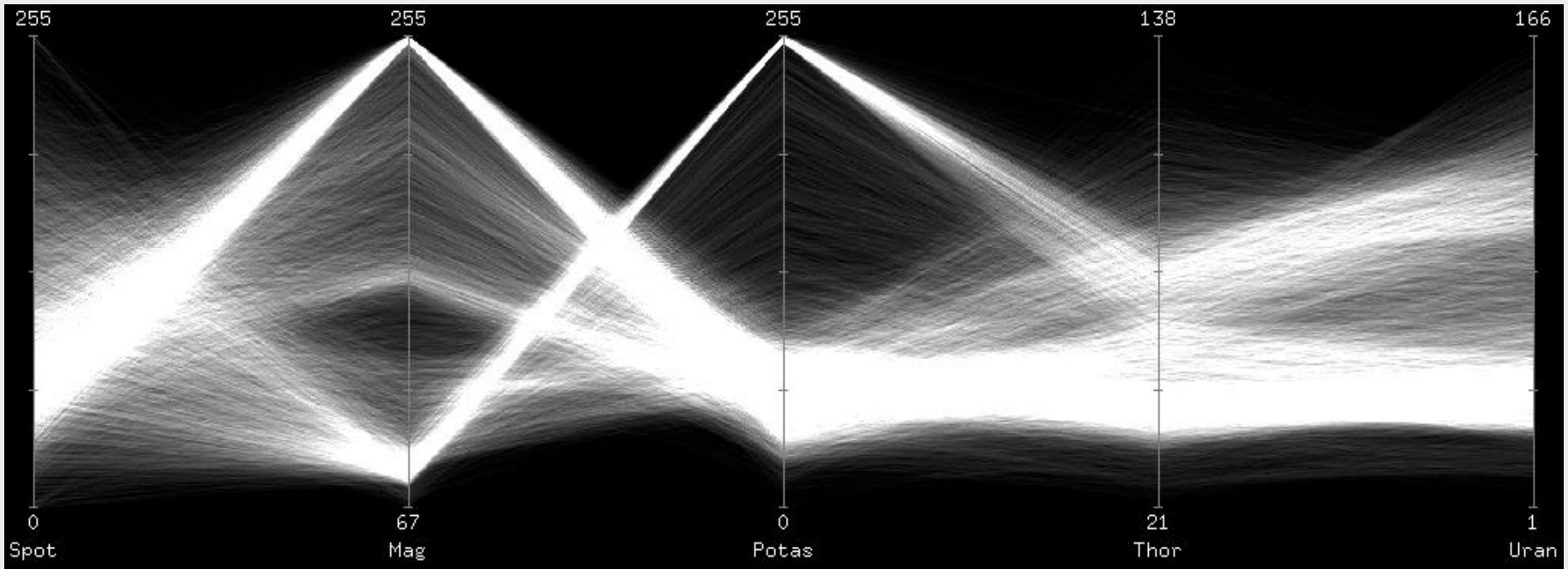


# PARALLEL COORDINATES PROPERTIES

## GRAPHICALLY INTENSE

One item → thousand pixels

Overplotting, low capacity



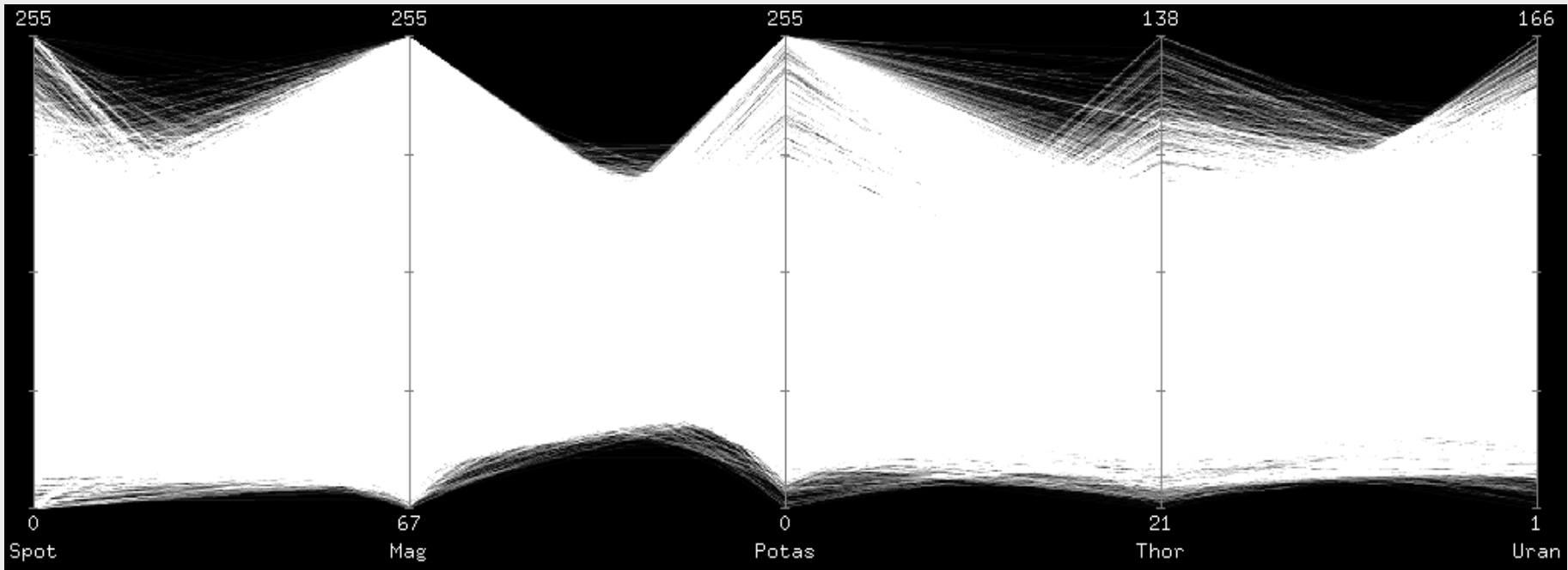
16.000 SAMPLES

# PARALLEL COORDINATES PROPERTIES

## GRAPHICALLY INTENSE

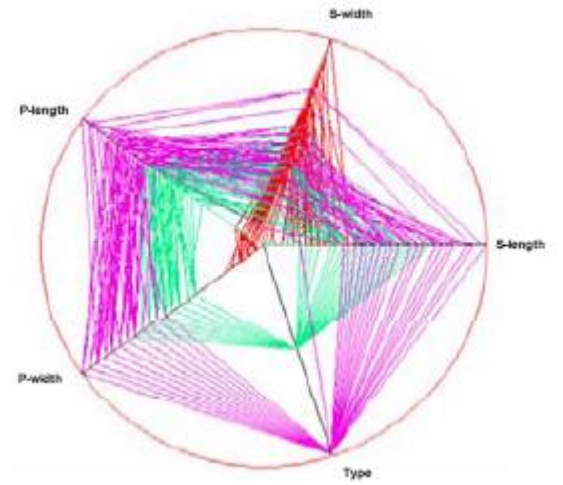
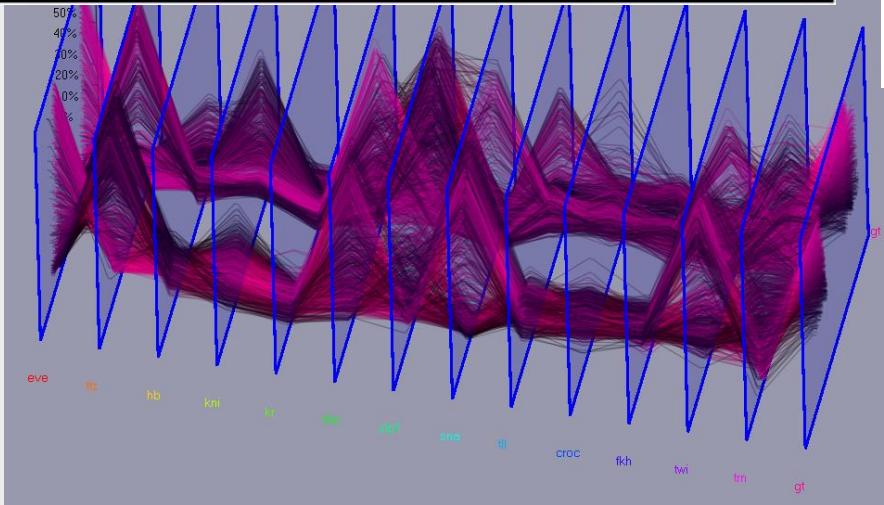
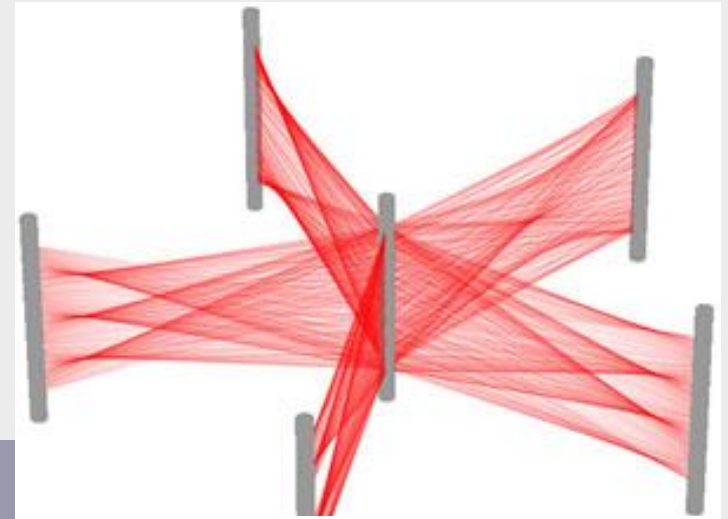
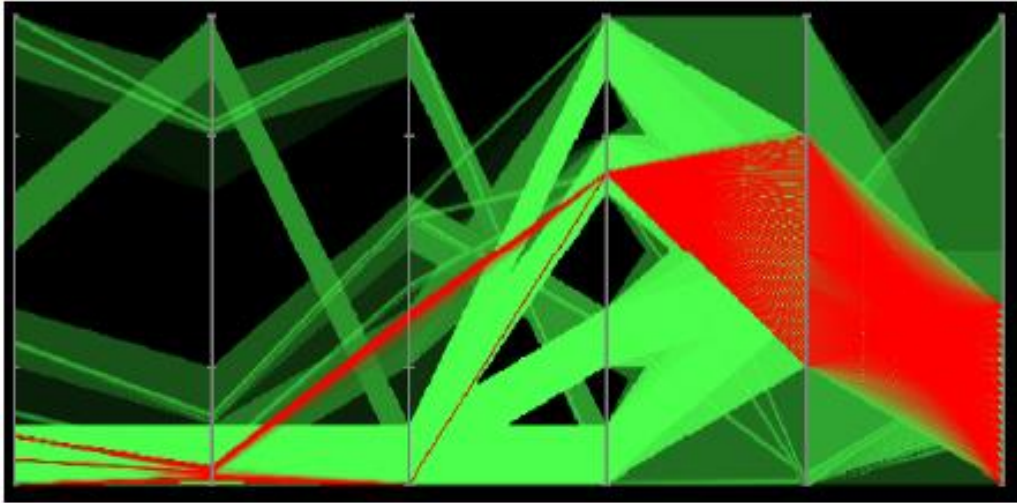
One item  $\rightarrow$  thousand pixels

Overplotting, low capacity



100.000 SAMPLES

# MODIFICATIONS



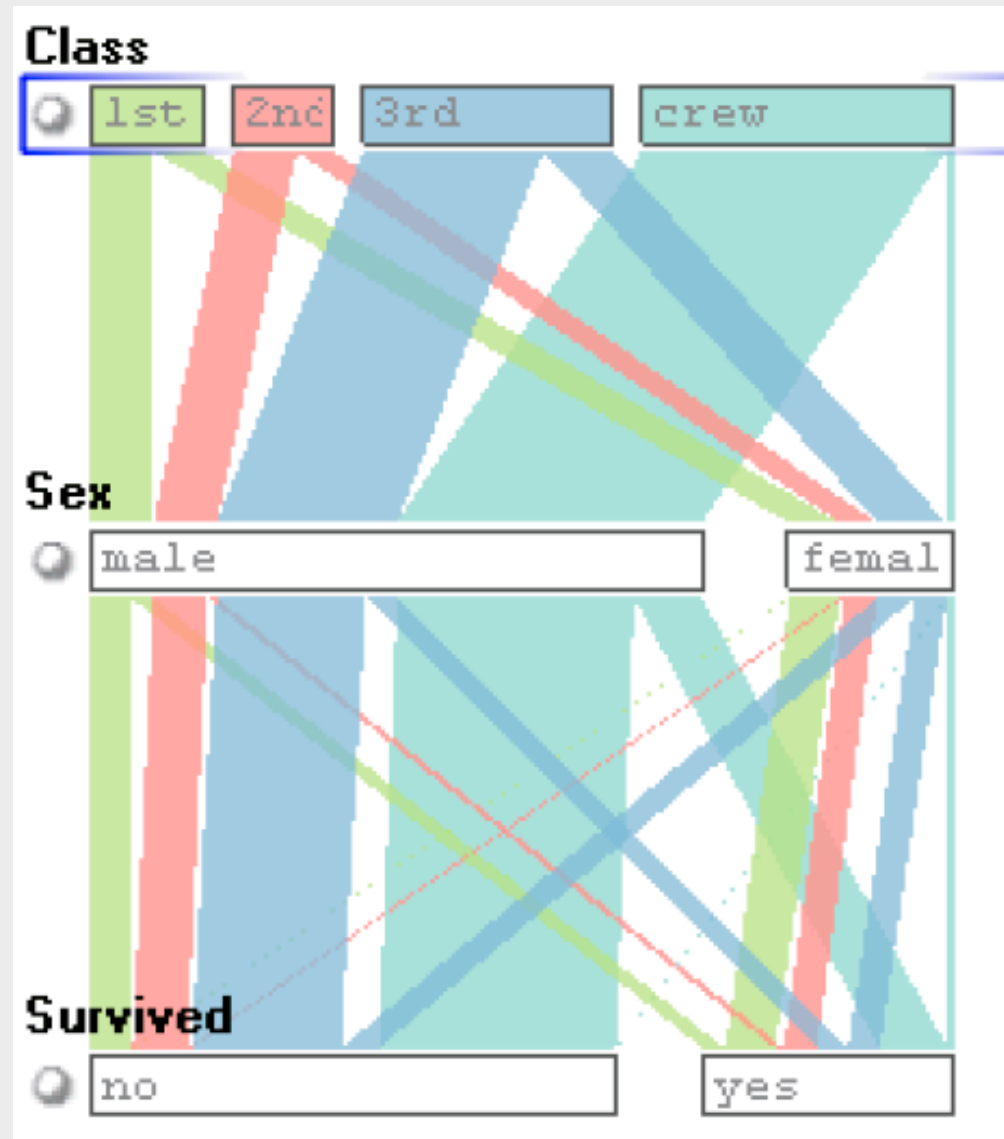
# PARALLEL SETS

CATEGORICAL DATA

PARALLEL AXES

COLORING BY ONE  
ATTRIBUTE

TITANIC DATA →



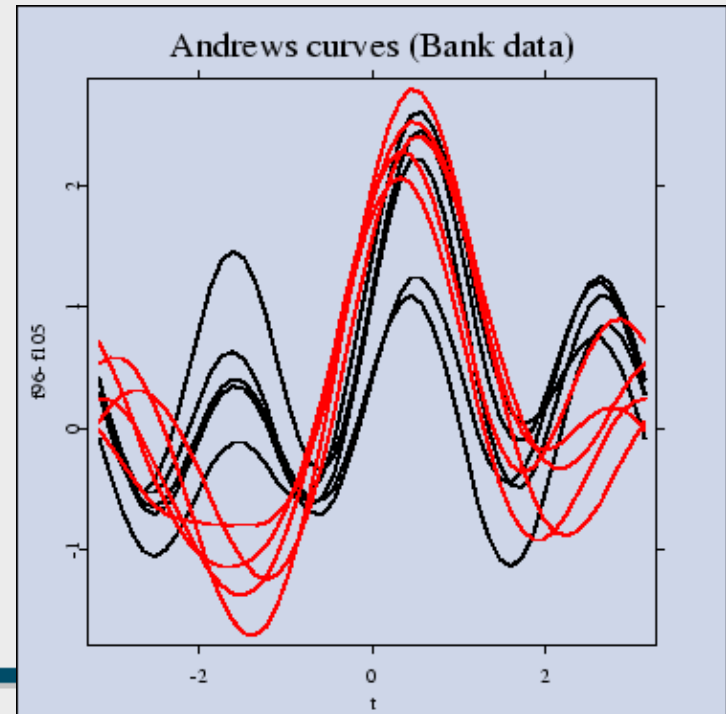
# ANDREWS PLOT

COORDINATE SYSTEM BASED ON  
ORTHOGONAL TRIGONOMETRIC FUNCTIONS  
(LIKE FOURIER)

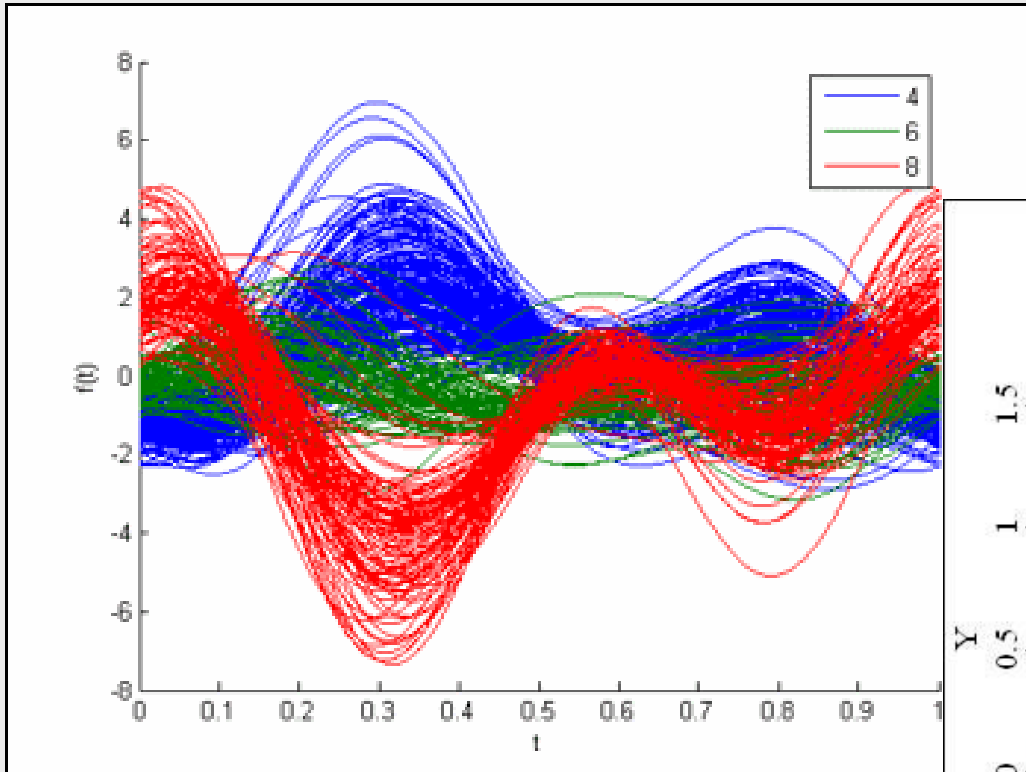
$$F_i(t) = X1_i/\text{SQRT}(2) + X2_i*\text{SIN}(t) + X3_i*\text{COS}(t) + X4_i*\text{SIN}(2t) + X5_i*\text{COS}(2t) + \dots$$

NO SEMANTICS IN X,Y

ORDER OF DIMENSIONS  
SUPERIMPORTANT

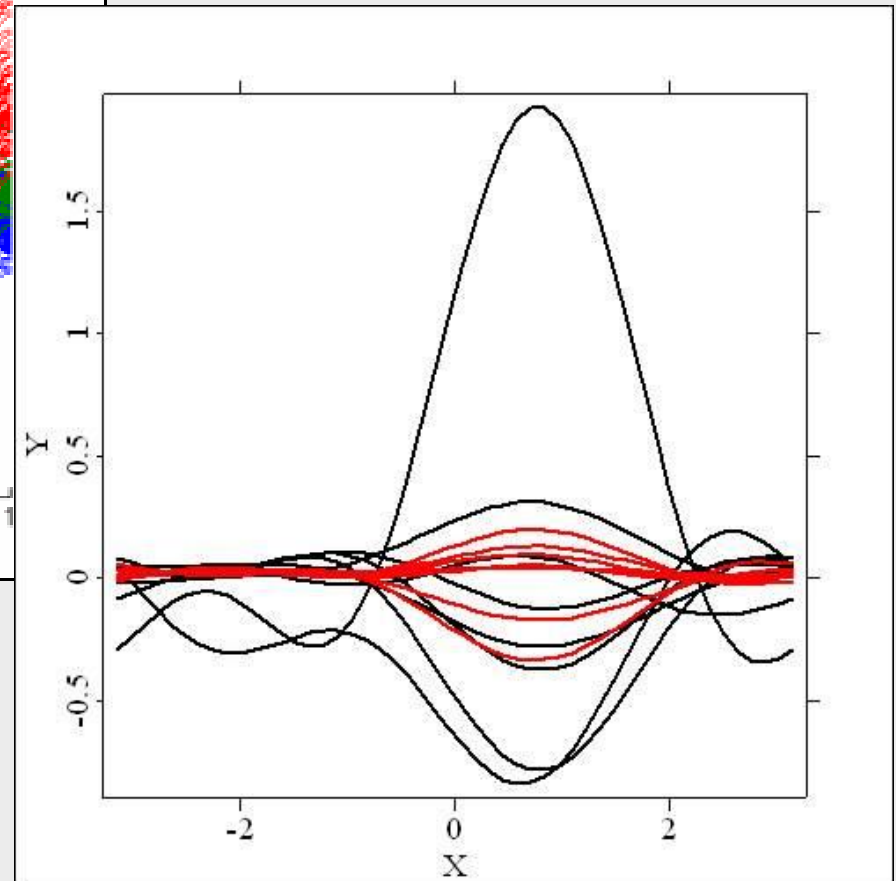


# ANDREWS PLOT



← CARS

STOCKS →



# SUMMARY

ORTHOGONAL COORDINATE SYSTEMS USE UP SPACE

RECONFIGURATION OF AXES OFFERS MORE DIMENSIONS

RECONFIGURED COORDINATES SYSTEMS MAY

Be non-intuitive (e.g. par.coordinates, andrews plot)

Require learning

Be abstract (e.g. trigonometric functions)

---

# TIME-DEPENDENT DATA

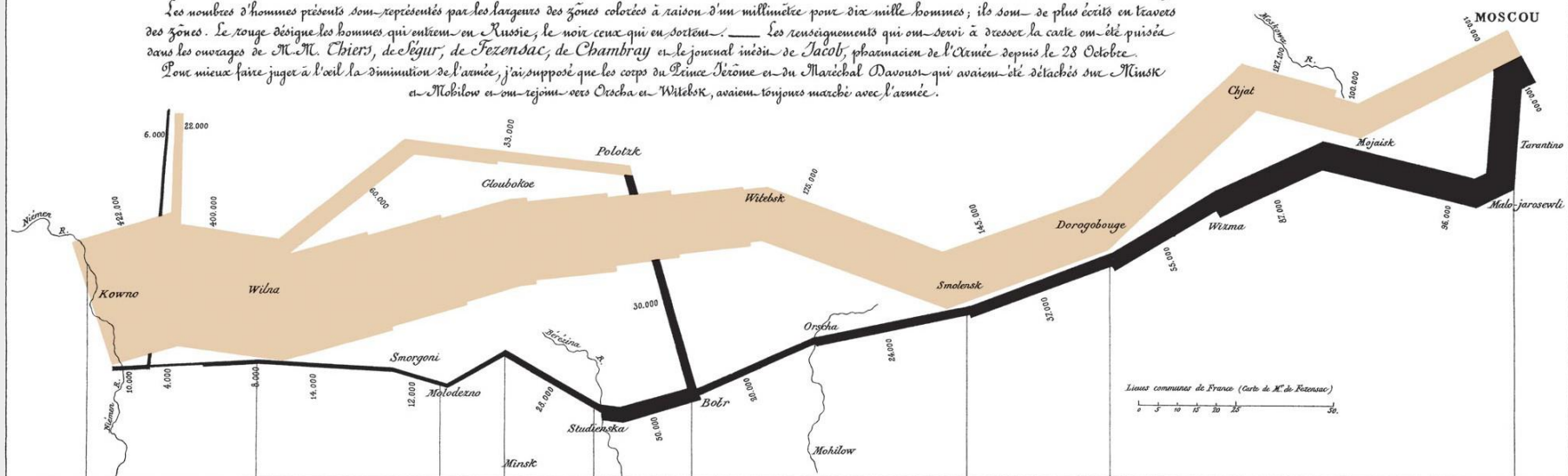


# HISTORY

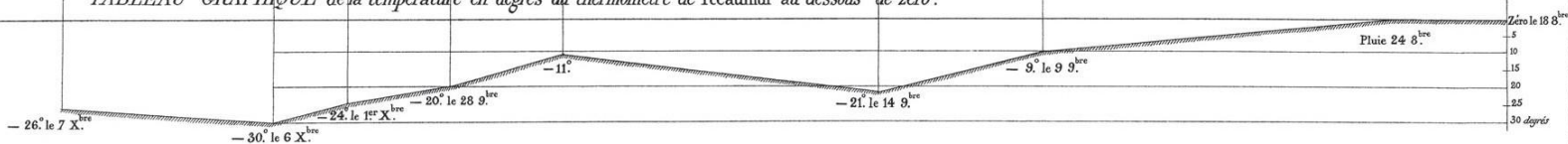
## CHARLES MINARD, 1869 NAPOLEON CAMPAIGN TO RUSSIA IN 1812

*Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813.*  
Dressée par M. MINARD, Inspecteur Général des Ponts et Chaussées en retraite. Paris, le 20 Novembre 1869.

Les nombres d'hommes présents sont représentés par les largeurs des zones colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en lettres dans des zones. Le rouge désigne les hommes qui ont été en Russie, le noir ceux qui en sont sortis. Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M. M. Chiers, de Légar, de Fozzard, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre. Pour mieux faire juger à l'œil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davout, qui avaient été détachés sur Minsk et Mabilow et qui rejoignirent Orscha et Witebsk, avaient toujours marché avec l'armée.



*TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.*



Les Cosaques passent au galop le Niémen gelé.

Auég. par Eugénie, 8. Par. 5<sup>me</sup> Marie St. O<sup>me</sup> à Paris.

Imp. Lit. Pagnier et Doucet.

# MOTIVATION

*"PREDICTION IS VERY DIFFICULT,  
ESPECIALLY ABOUT THE FUTURE"*

UNDERSTANDING TEMPORAL RELATIONS  
HELPS PREDICT FUTURE

DETECTING EVENTS HELPS LEARN FROM THE  
PAST AND FIND RELATIONS

*References:* Aigner et al. :  
Visualizing Time-Oriented Data - A Systematic View

---

# STRUCTURE OF TIME AXIS

## TIME POINTS VS. TIME INTERVALS

### GRANULARITY

seconds, weeks, ...

### LINEAR TIME

E.g. average income over last 50 years

### CYCLIC TIME

E.g. website visitors over a week

### BRANCHING TIME

E.g. project development

---

# DIMENSIONALITY

## SINGLE DIMENSION

E.g. ocean level over past 50 years

## MULTIPLE DIMENSIONS

Multiple dimensions in different time points

E.g. temperature, humidity, wind speed measured at different (not the same) time

## MULTIDIMENSIONAL

Complete multidimensional snapshots at different time points

E.g. credit card transactions

---

# ENTITIES

## SINGLE ENTITY

Various temporal aspects of 1 entity  
E.g. health monitor of a patient

## MULTIPLE ENTITIES

Track development of each entity over time  
e.g. GDP of European countries over years

## UNKNOWN ENTITIES

No entity match between timesteps  
E.g. anonymous censuses from different years

---

# UNCERTAINTY

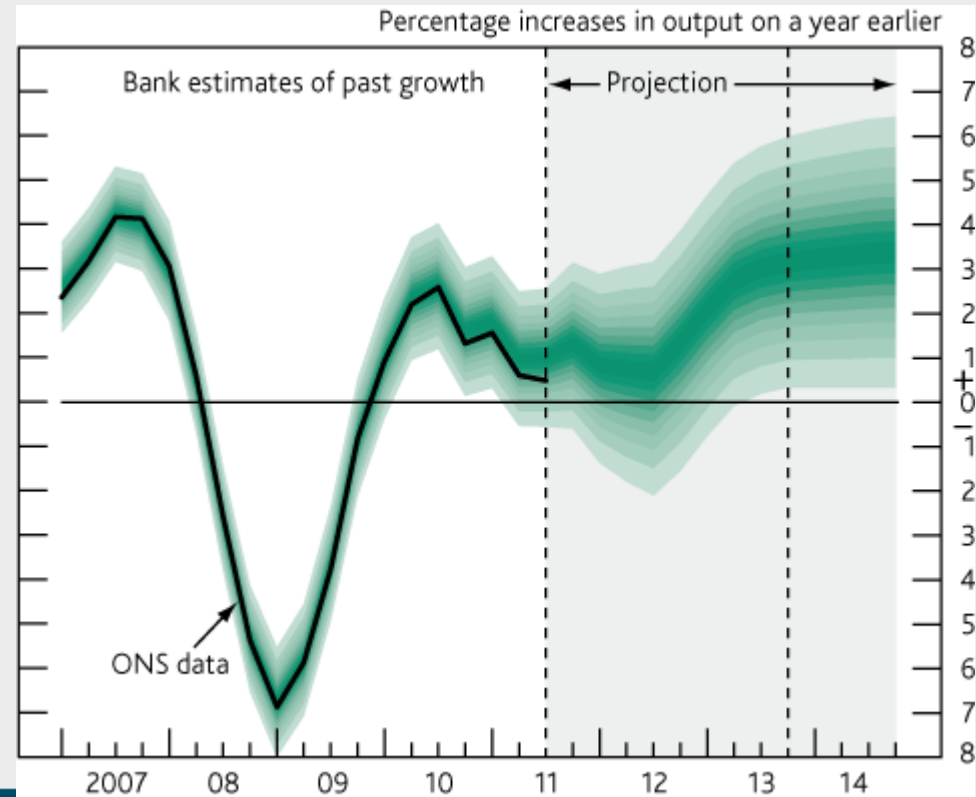
USUALLY ABOUT FUTURE DATA

LEVEL OF UNCERTAINTY

PROBABILITY (0..1)

VARIANCE

DENSITY  
DISTRIBUTION



# STREAMED DATA

FREQUENCY AND DYNAMICS

HOW OFTEN DATA CHANGES?

WHAT IS THE VARIATION OF DATA?

WHEN WAS THE LAST UPDATE

E.G. TRAFFIC INFORMATION, WEATHER  
INFORMATION

---

# GOALS OF TEMPORAL VISUALIZATION

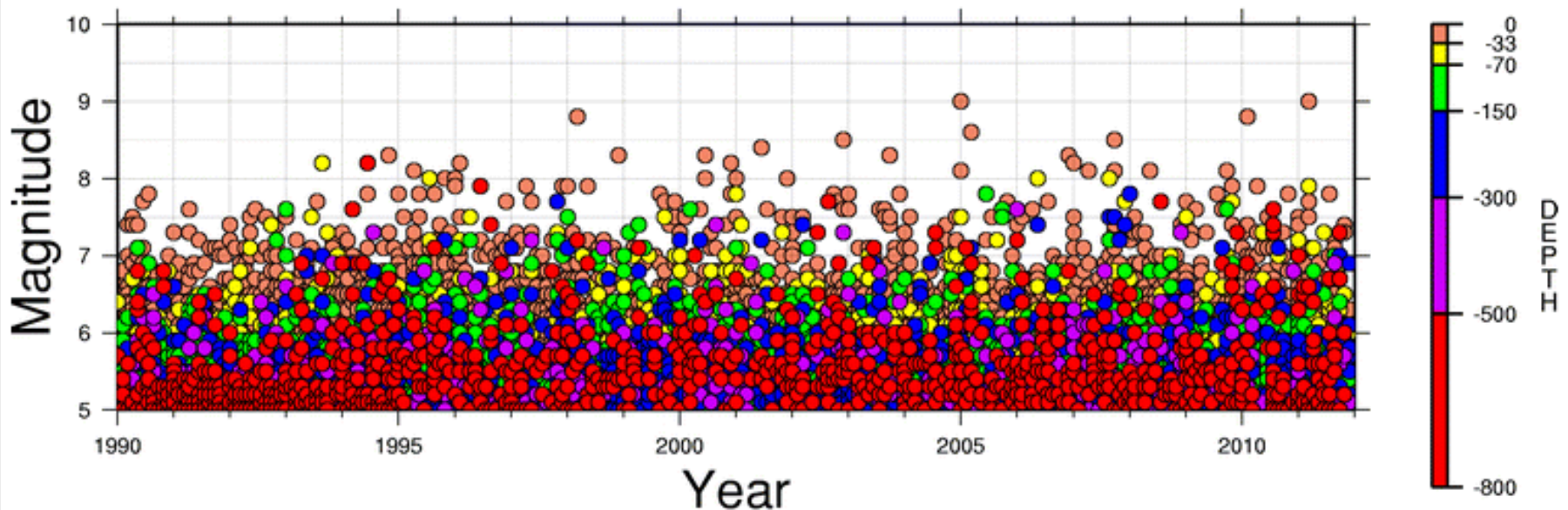


# COMPARE DATA IN DIFFERENT TIMESTEPS

## DIFFERENCES BETWEEN SPECIFIC TIME STEPS

### Earthquakes Located by the NEIC

### Magnitude 5 and Greater



Mon Nov 21 03:50:00 MST 2011

USGS National Earthquake Information Center

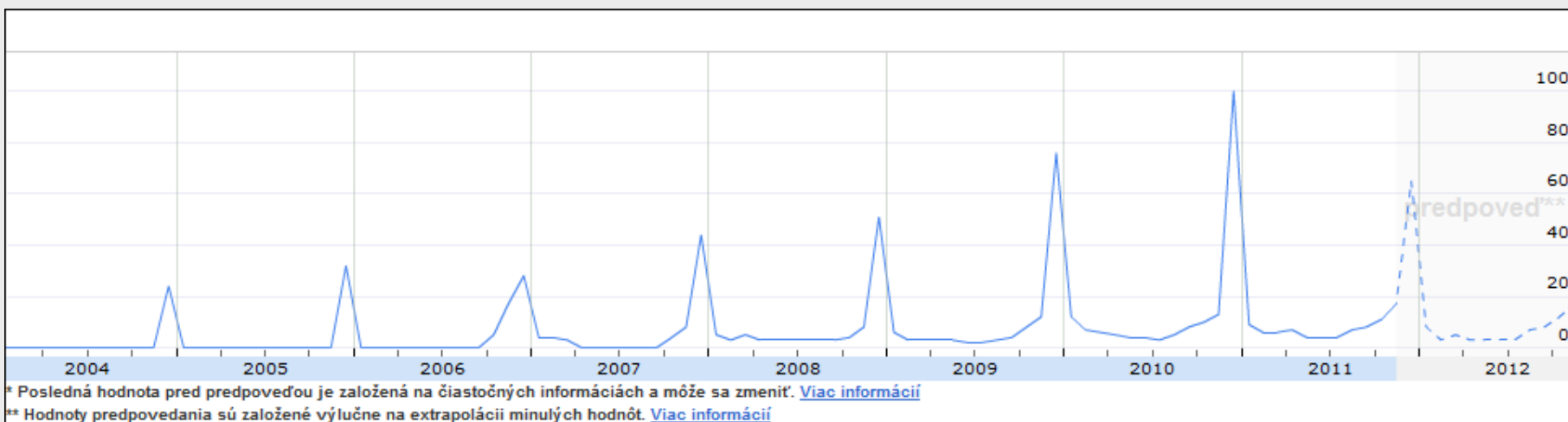
# DETECT EVENTS

EVENT =  
WHEN A SIGNIFICANT CHANGE HAPPENS



# DETECT PATTERNS IN 1 ATTRIBUTE

INCREASE, DECREASE, REPEAT, JUMP, DROP, ...



# DETECT LAYERED PATTERNS

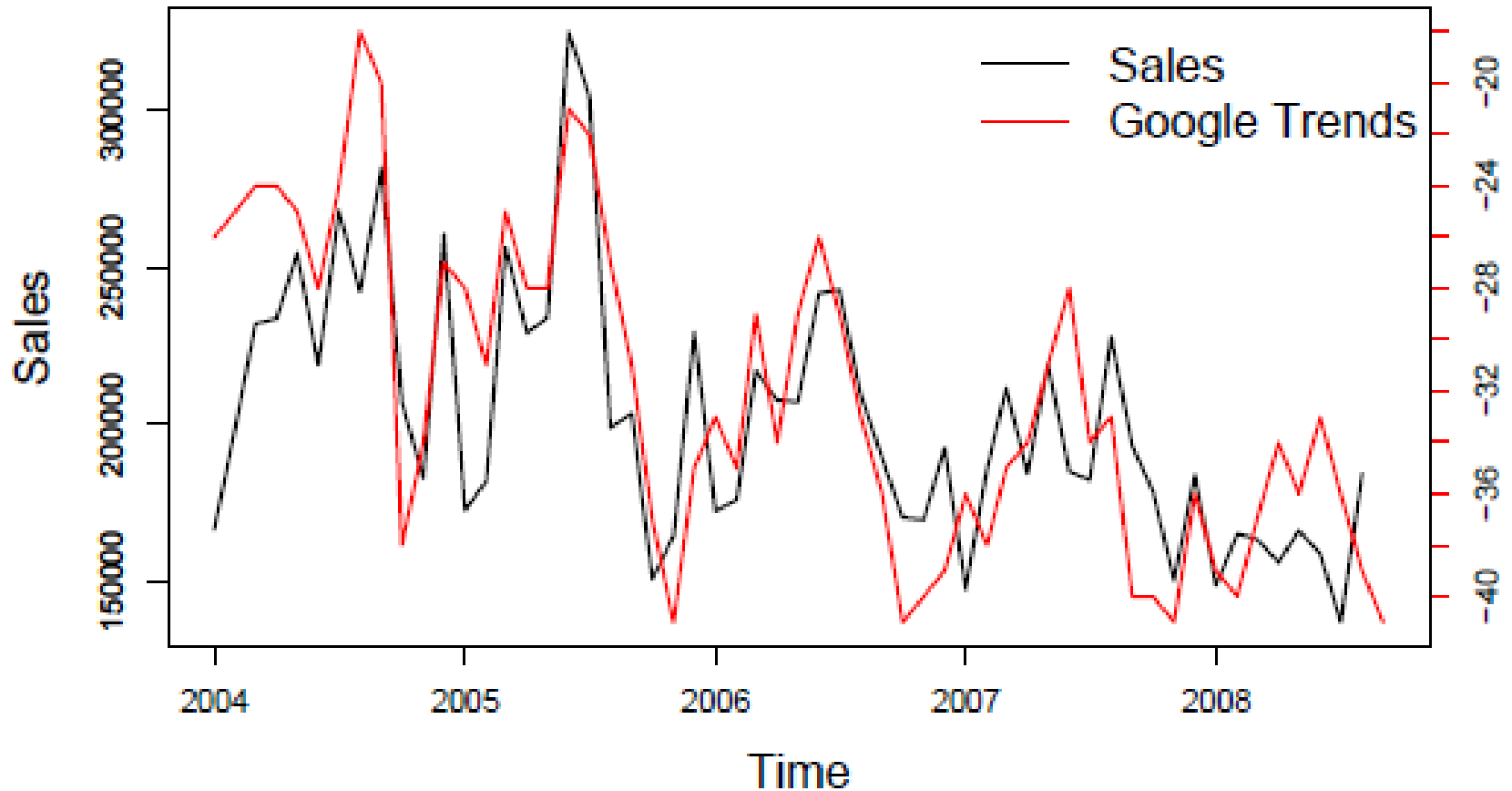
## LOCAL + GLOBAL TRENDS



# DETECT ATTRIBUTE RELATIONS

ARE TWO VARIABLES RELATED IN TIME?

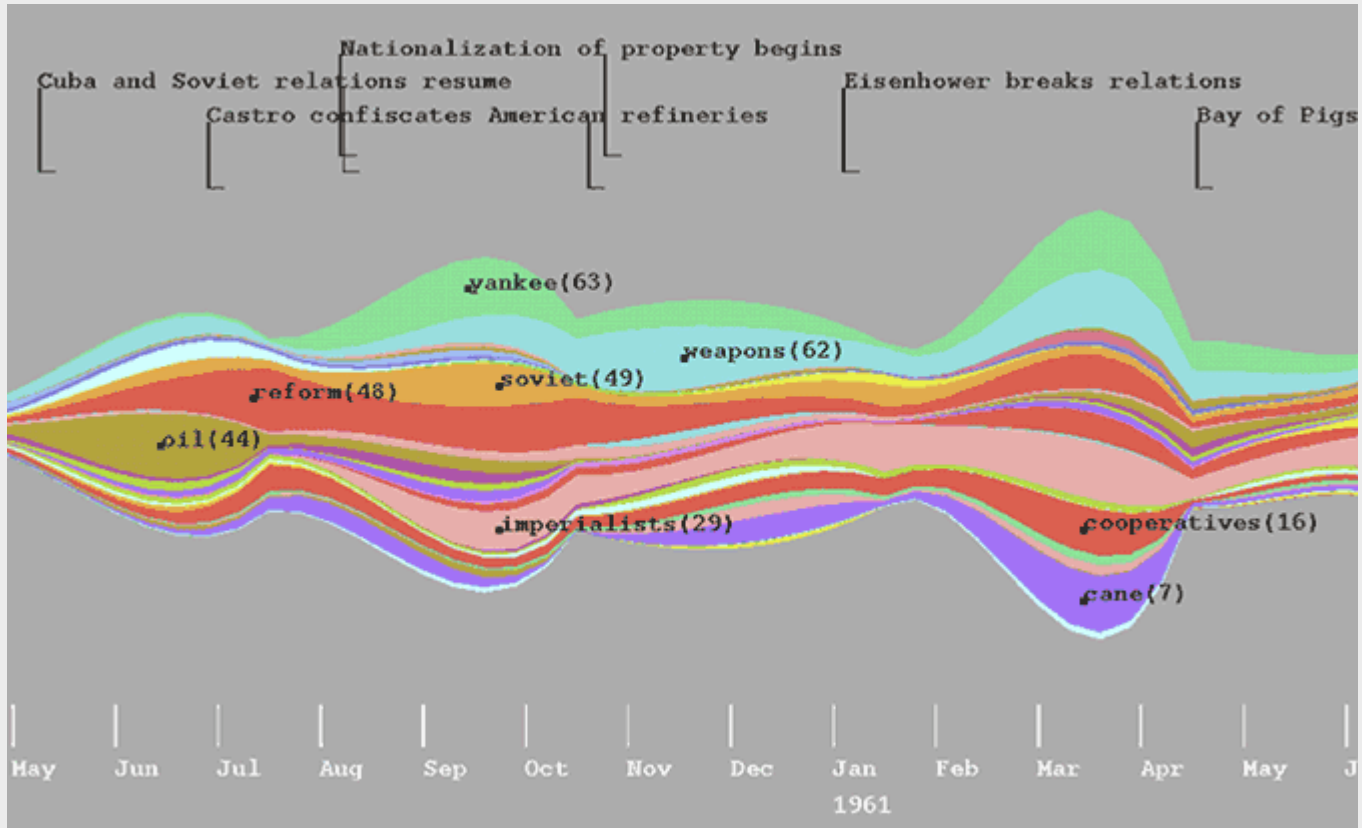
## Chevrolet



# EXAMPLE TECHNIQUES

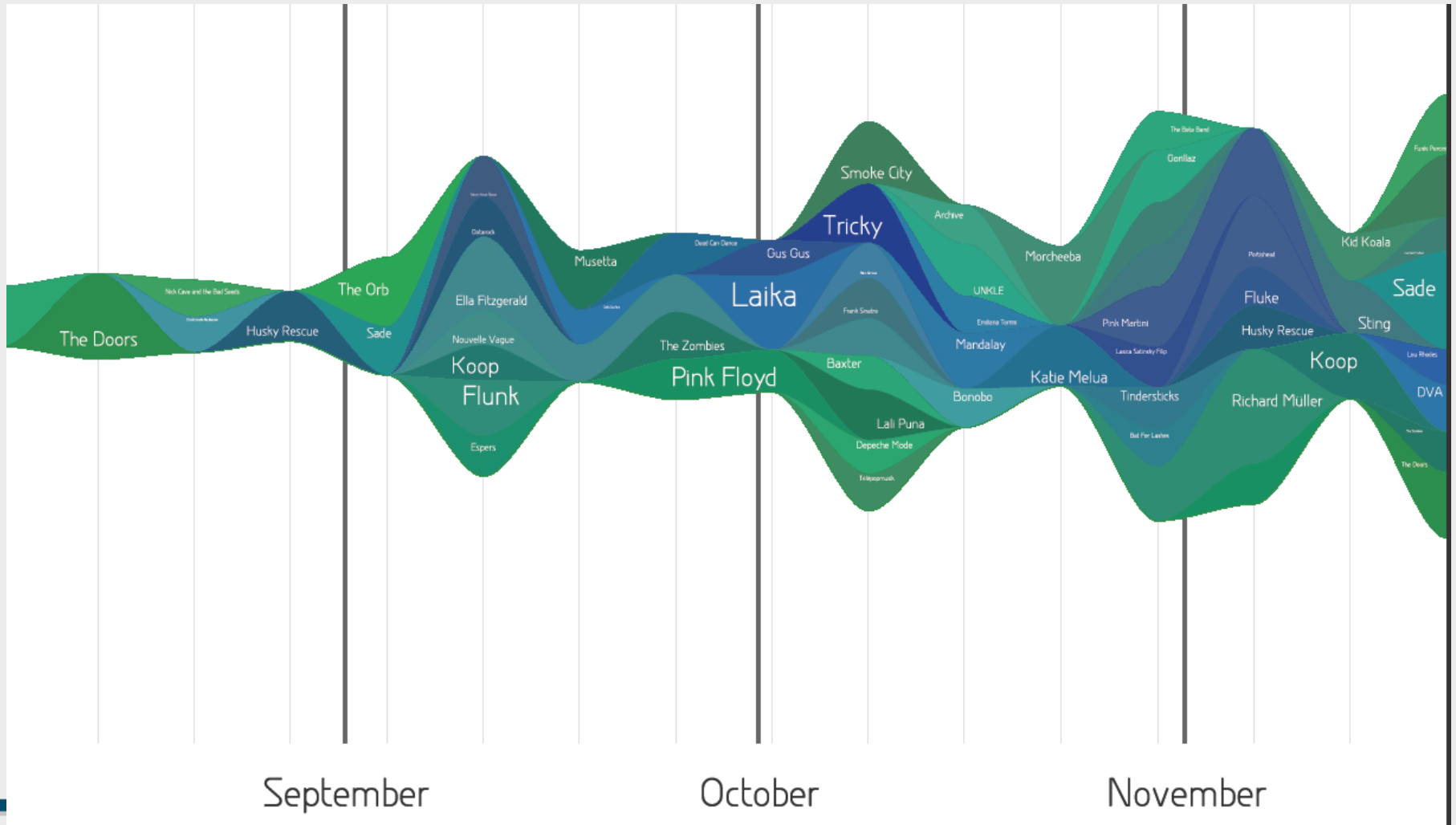
# THEME RIVER

HISTOGRAM IN DIFFERENT TIME STEPS.  
SPLINE INTERPOLATION OF VALUES => AREAS



# THEME RIVER CLONES

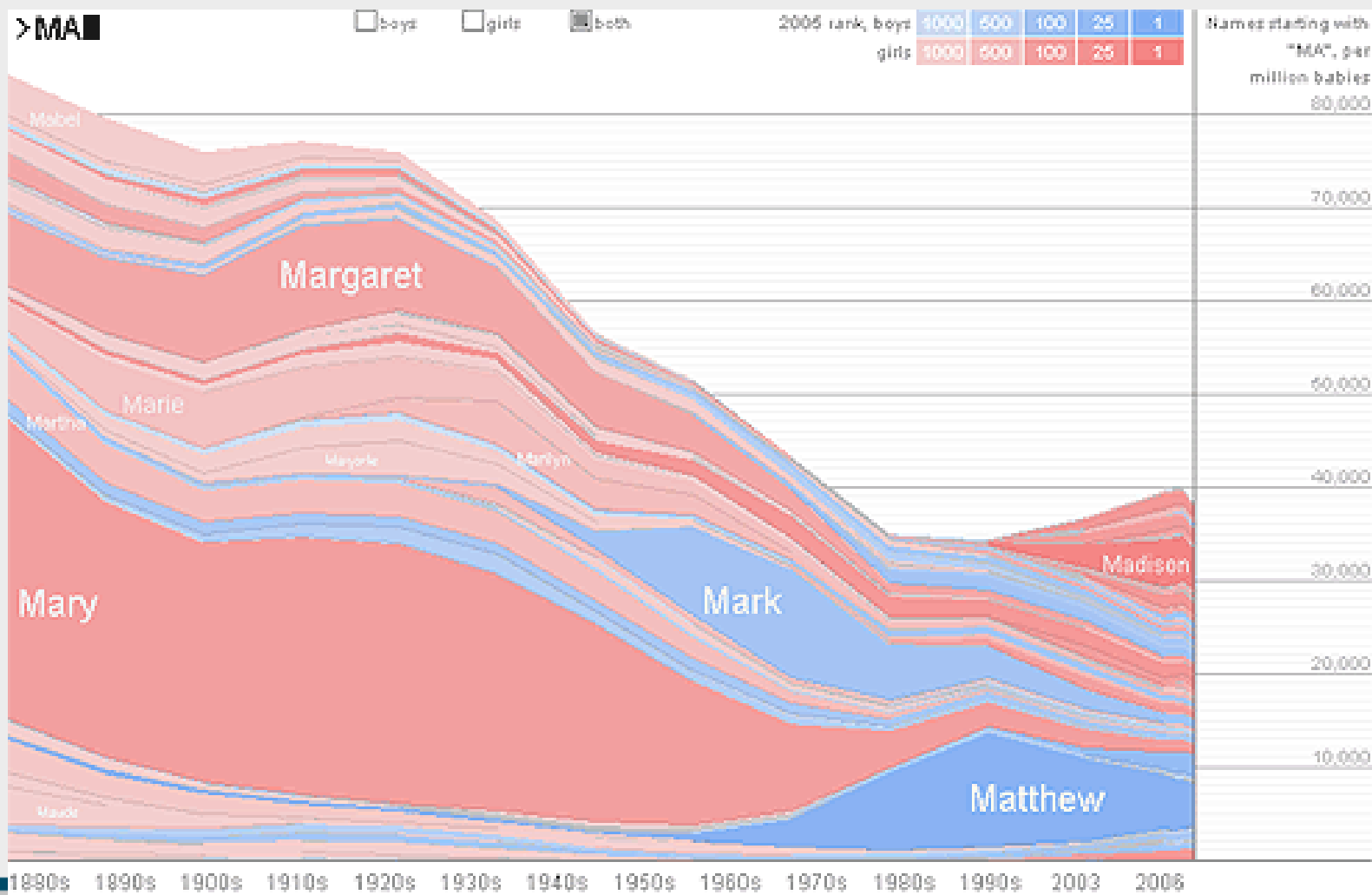
<http://lastgraph.aeracode.org/>





# THEME RIVER CLONES

<http://babynamewizard.com/voyager>

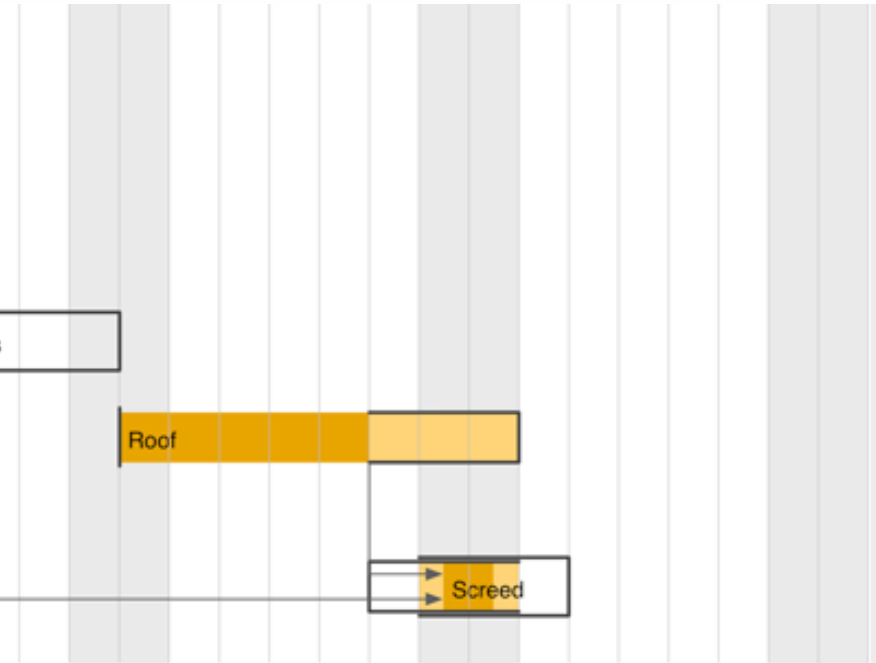
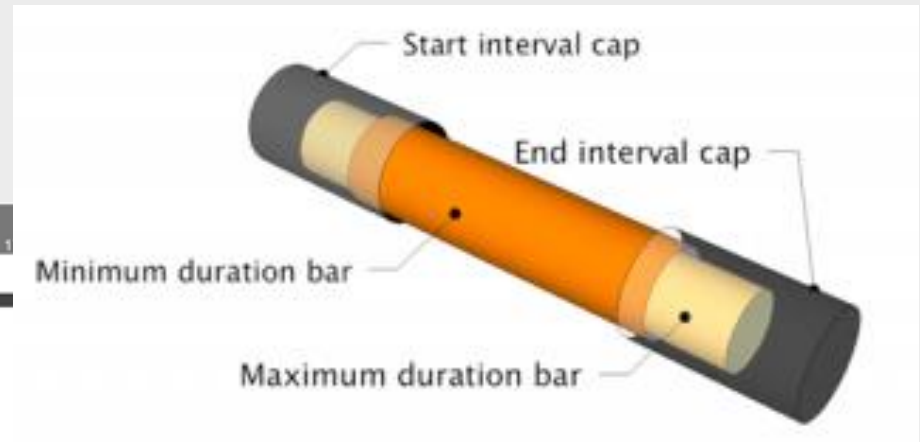
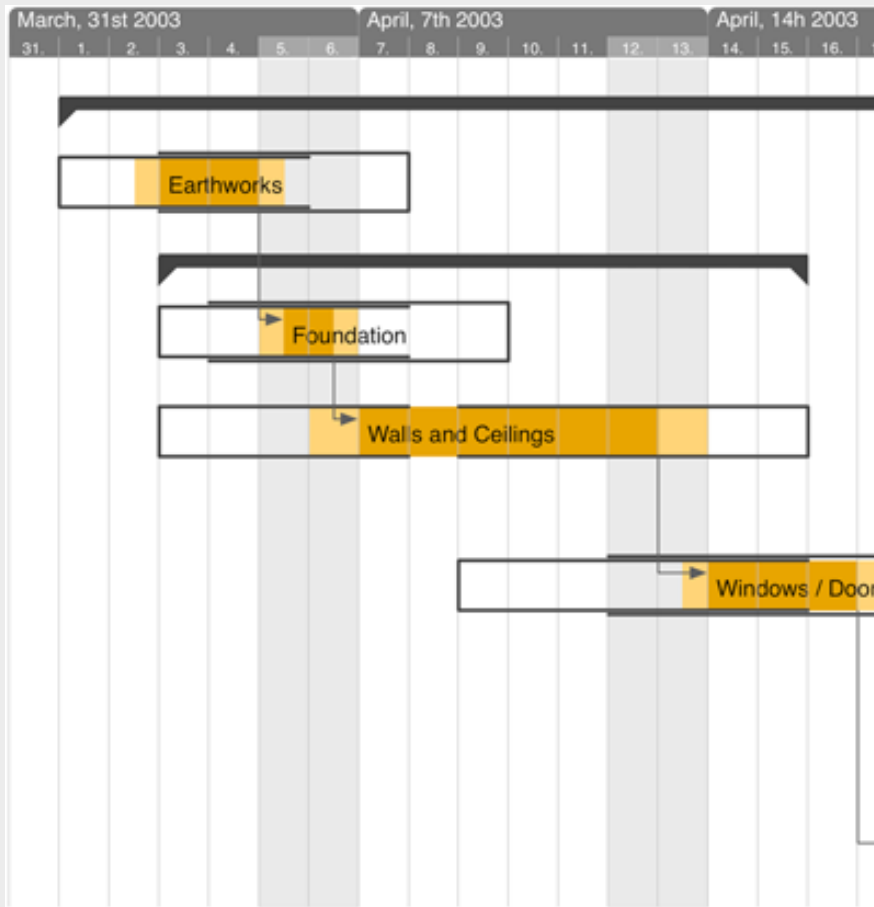


# GANTT CHART & MODIFICATIONS

	11.03	12.03	1.04	2.04	3.04	4.04	5.04	6.04
<b>Preparation and Planning</b>								
Develop project proposal								
Approve project proposal								
Recruit project team								
<b>Development and Test</b>								
Specify detail requirements								
Develop prototype								
Approve prototype								
Develop beta version								
Test beta version								
Apply final corrections								
Approve final version								
<b>Implementation</b>								
Train users								
Roll-out final version								

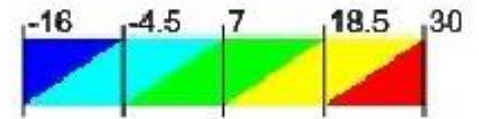
# GANTT CHART MODIFICATIONS

## PLANNING LINES



# TWO-TONE PSEUDO COLORING

Priemerná denná teplota



2007



2006



2005



2004



2003



2002



2001



2000



# SPIRAL GRAPH

TIME AXIS AS A SPIRAL.  
CYCLE LENGTH IS A PARAMETER  
=> PERIODICITY CAN BE REVEALED

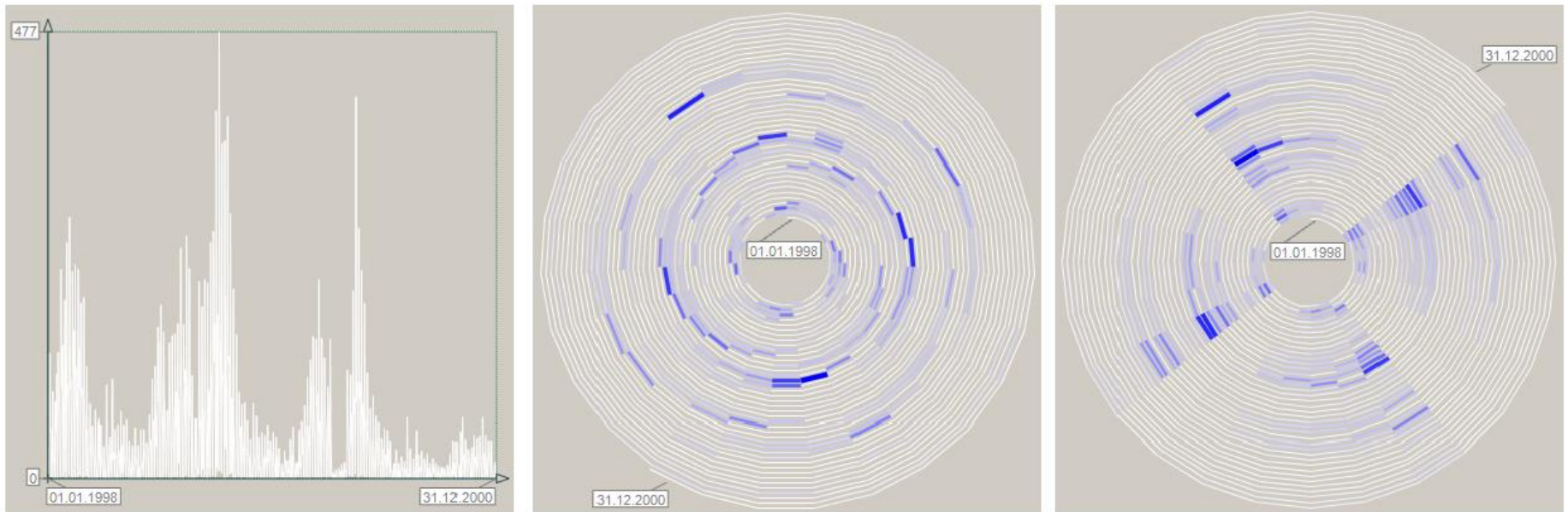


Fig. 1. Different visual representations of a time-oriented dataset describing the number of influenza cases over a period of three years – left: Time series plot (periodic pattern is difficult to discern), center: SpiralGraph encoding 27 days per cycle (improperly parameterized – periodic pattern is hard to see), right: SpiralGraph encoding 28 days per cycle (properly parameterized – periodic pattern stands out).

# BRANCHING TIME

www.digibarn.com/collections/posters/tongues/

## Mother Tongues

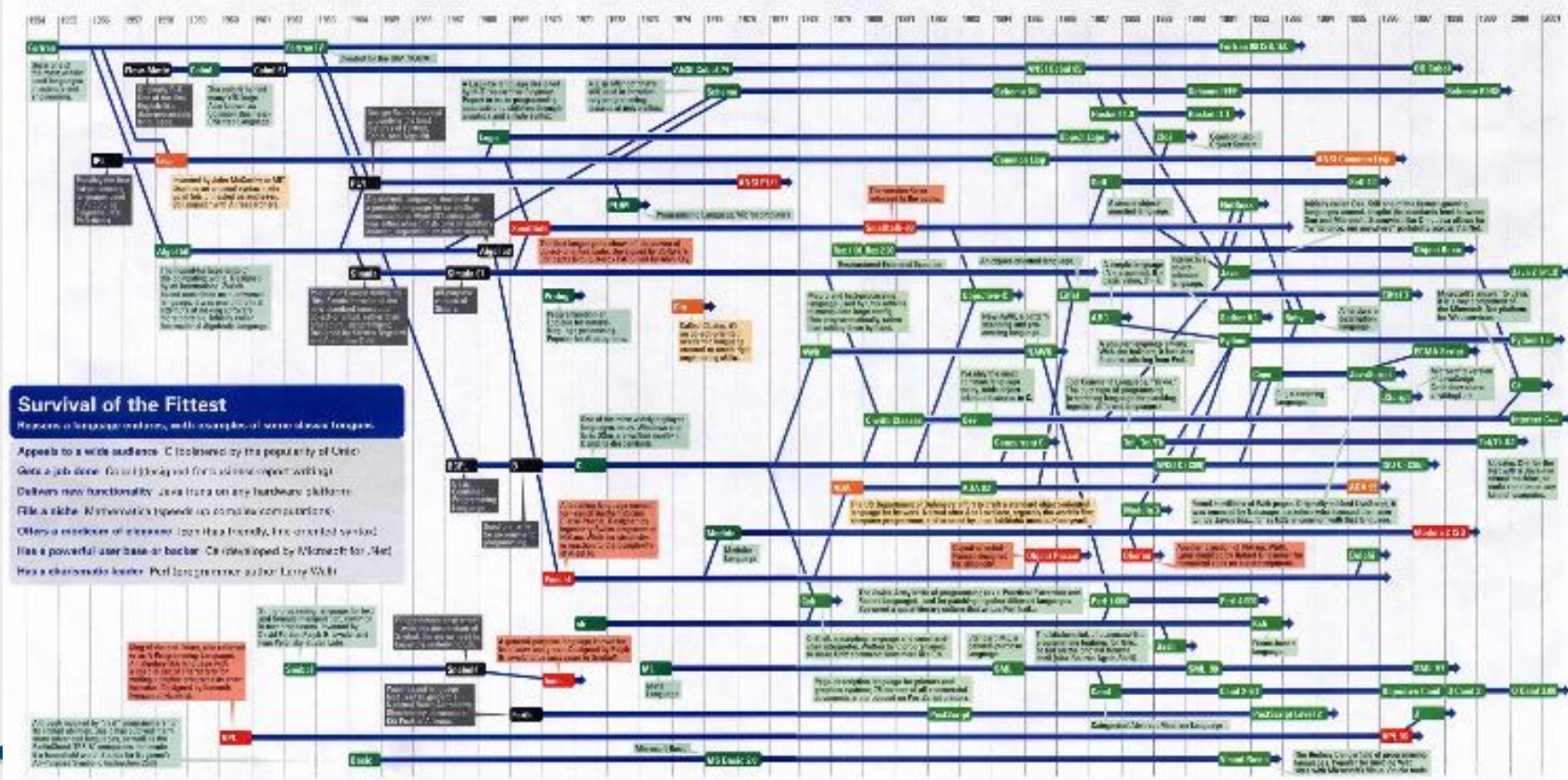
Tracing the roots of computer languages through the ages

Just like half of the world's spoken tongues, most of the 2,000-plus computer programming languages are either developed or distinct. As powerful as C++, Visual Basic, Pascal, Java, and other modern source codes can make our systems, hundreds of older languages are running out of life. An old collection of languages—almost forgotten languages, if you will—almost none, or at least dormant, in the lingo of classic software. They're seeking the jobs of \$ million developers in search of codes that'll flourish in a nearly forgotten lingua franca. Among the most endearing are Ada, APL, Fortran, the predecessor of C, Lisp, Oberon, Smalltalk, and Simula.

Code writer Grady Booch, National Software's chief scientist, is working with the Computer History Museum in Silicon Valley to record and, in some cases, resurrect languages by writing new compilers to run on changing hardware and stack the books. Why bother? "They tell us about the state of software practice, the minds of their inventors, and the technical, social, and economic forces that shaped history in the days," Booch explains. "They'll provide the raw material for software education, traditions, and innovations to learn what worked, what was brilliant, and what was an unfortunate. Here's a peek at the historical branches of programming's family tree. For a nearly exhaustive rundown, check out the Language List at [www.cwi.nl/~wim/lectures/teaching/and/andlect/lect1.html](http://www.cwi.nl/~wim/lectures/teaching/and/andlect/lect1.html) - Michael Meekins

**Key**

- 1954 Year introduced
- Active descendants of users
- Preserved, mostly of some other, computer scientist
- Got original usage (copying old)
- Derivative became a derivative of another derivative
- Language evolved

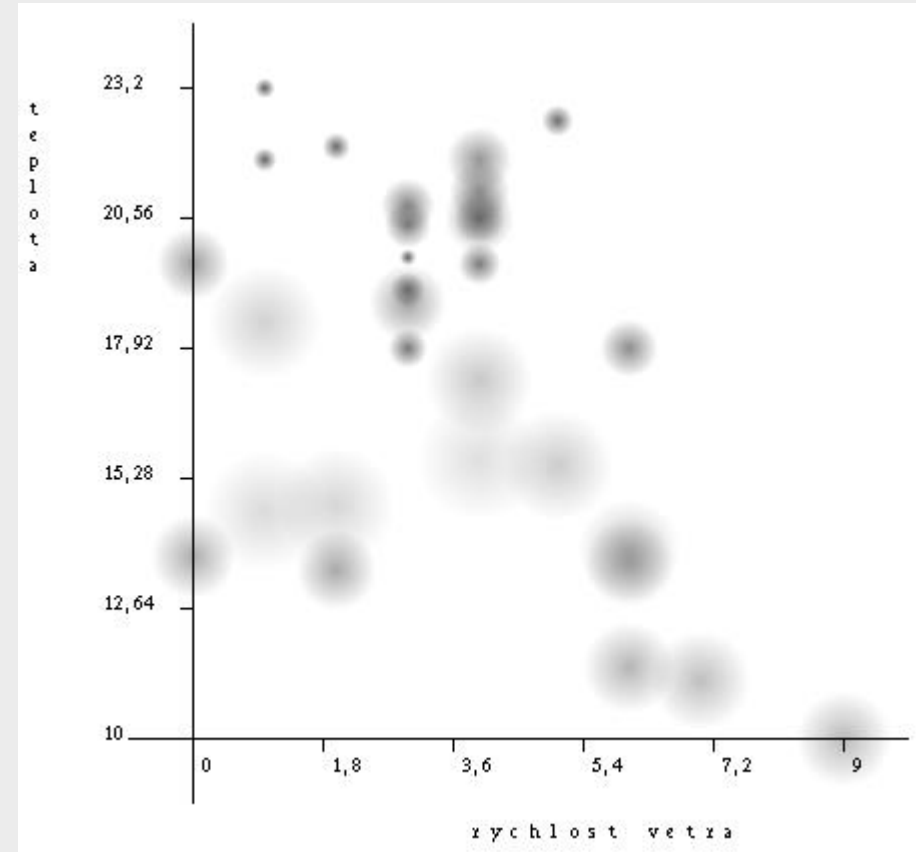
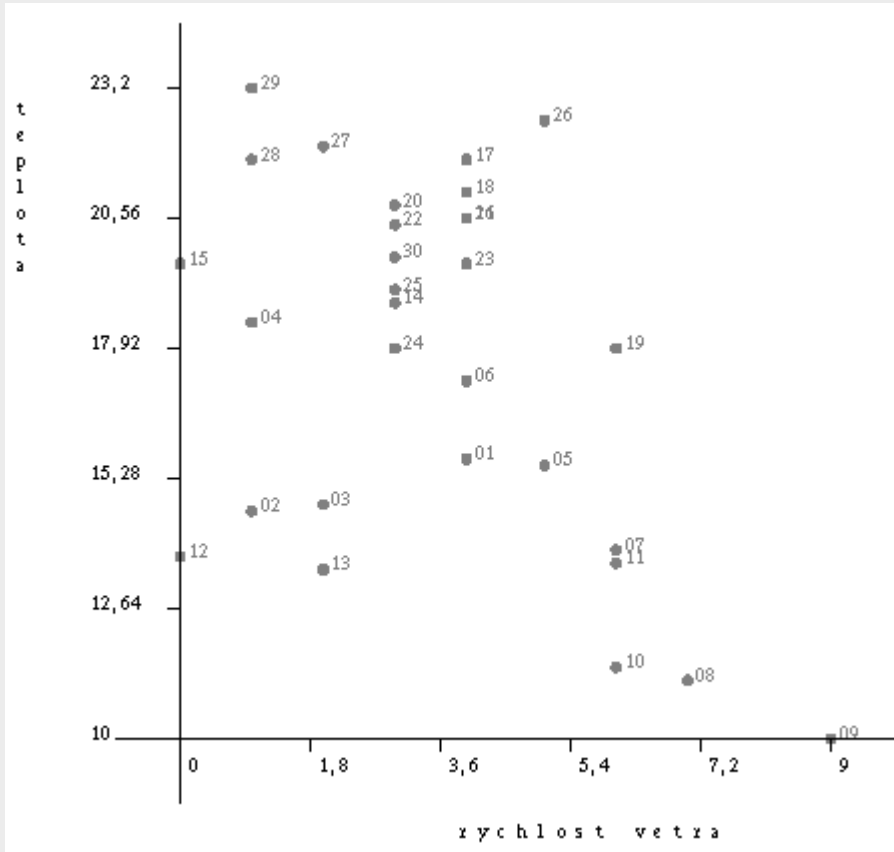


**Survival of the Fittest**  
Reasons a language endures, with examples of some classic tongues.

- Appeals to a wide audience: C (inspired by the popularity of Unix)
- Gets a job done: Fortran (and Fortran derivatives) expert writing
- Delivers new functionality: Java (runs on any hardware platform)
- Fills a niche: Ada (military's speeded-up digital computers)
- Offers a measure of elegance: Fortran (friendly, first compiled version)
- Has a powerful user base or backing: C++ (developed by Bjarne Stroustrup)
- Has a charismatic leader: Perl (programming author Larry Wall)

# TIME-DEPENDENT SCATTERPLOTS

## INK STAIN METAPHOR



# SUMMARY

## DATA TYPES

One or multiple attributes over time

One or multiple (or no known) entities over time

## VISUALIZATION OF TIME-DEPENDENT DATA

Temporal trends (rise, decline, periodicity)

Attribute correlation over time (helps prediction)

Event detection (outlier in temporal domain)

---