The Mathematical Foundation of Analytic Visualizations

Leland Wilkinson
SYSTAT
University of Illinois at Chicago (Computer Science)
Northwestern University (Statistics)

FODAVA Distinguished Lecture
Georgia Tech
April 2, 2010

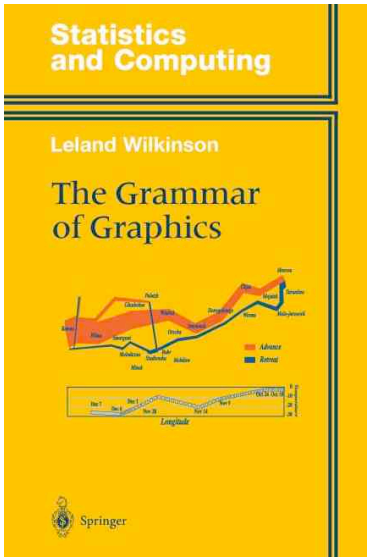
SYSTAT

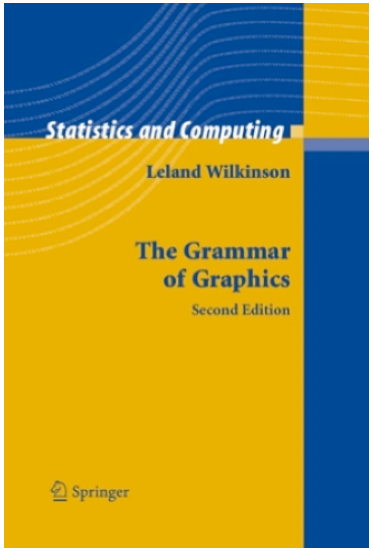
Outline

- A formal graphics language for analytics
  - An intelligent analytic system requires a language
  - The Grammar of Graphics (GoG)
- Aspects of a formal graphics language
  - Simple
  - Expressive
  - Coherent
  - Meaningful
- Intelligent graphics systems based on GoG

The Grammar of Graphics (1999, 2005)

- Show programmers how to design and implement statistical graphics
- Reveal the mathematical foundation of statistical graphics.

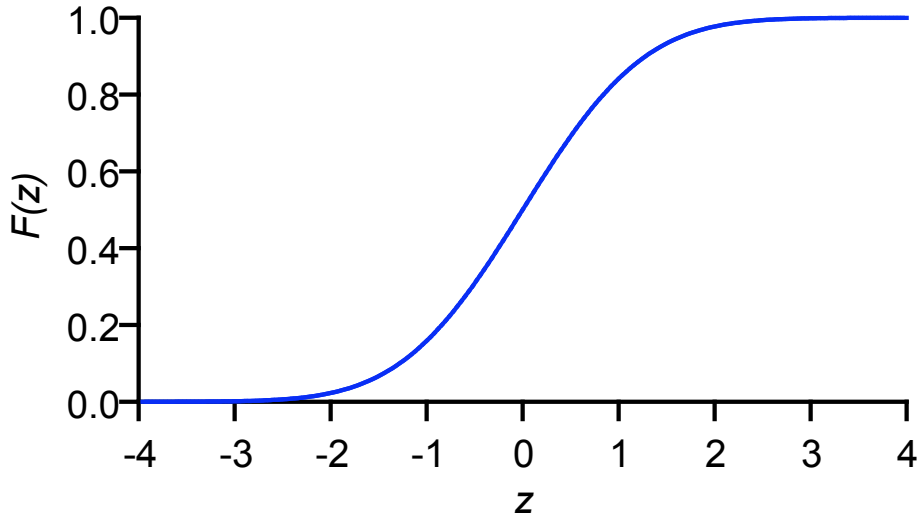




SYSTAT

The GoG Rule

- A statistical graphic is a representation of the graph of a function.
- The graph of a function is a subset of the product of its domain and codomain.
- The graphic representing $F(z) = \Phi(z)$ here is blue.
- The rest is annotation.



SYSTAT

What is a Graphic?

Graph

$G = \{(x, f(x)) : x \in R \text{ and } f(x) = e^{-x^2}\}$

Frame

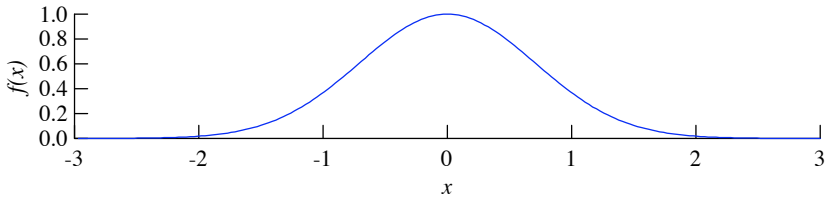
$F = [-3, 3] \times [0, 1]$

Aesthetic

$A : x \mapsto x_{position}, f(x) \mapsto y_{position}$

Graphic

$G_A = A(F \cap G)$

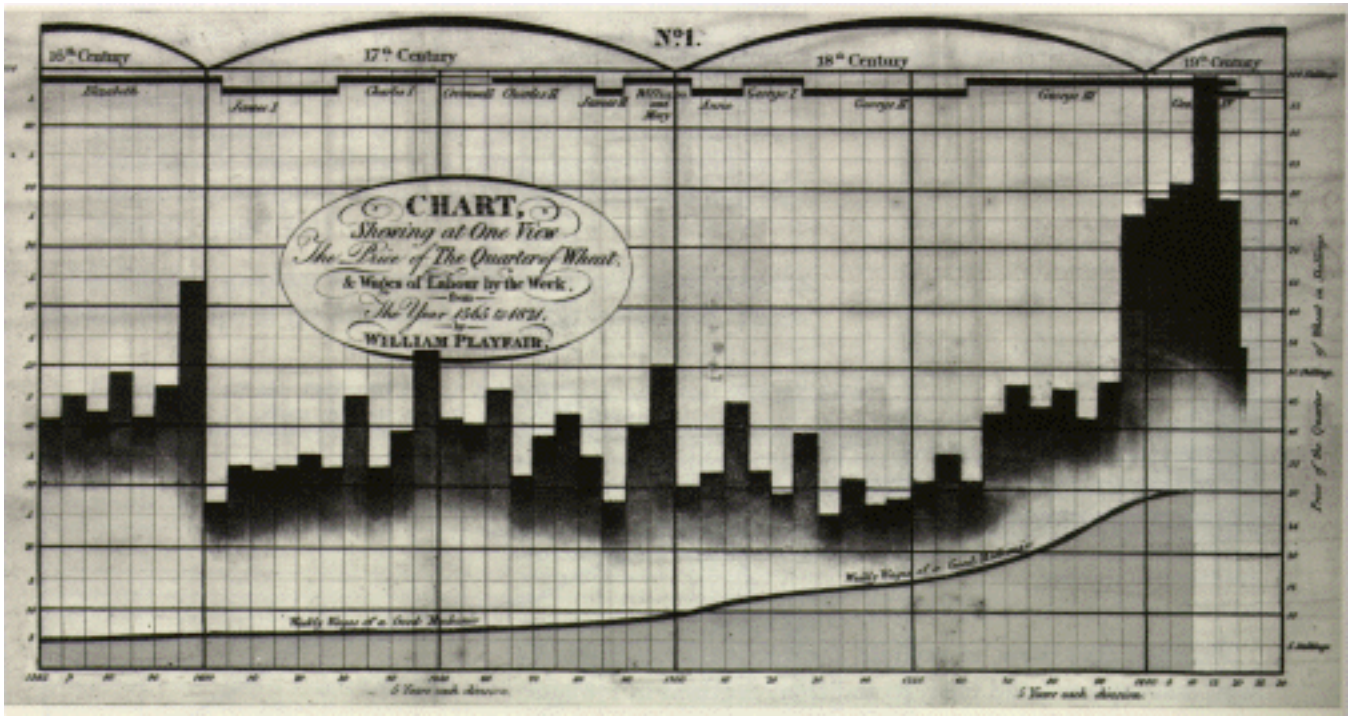


SYSTAT

The same rule applies to this graphic

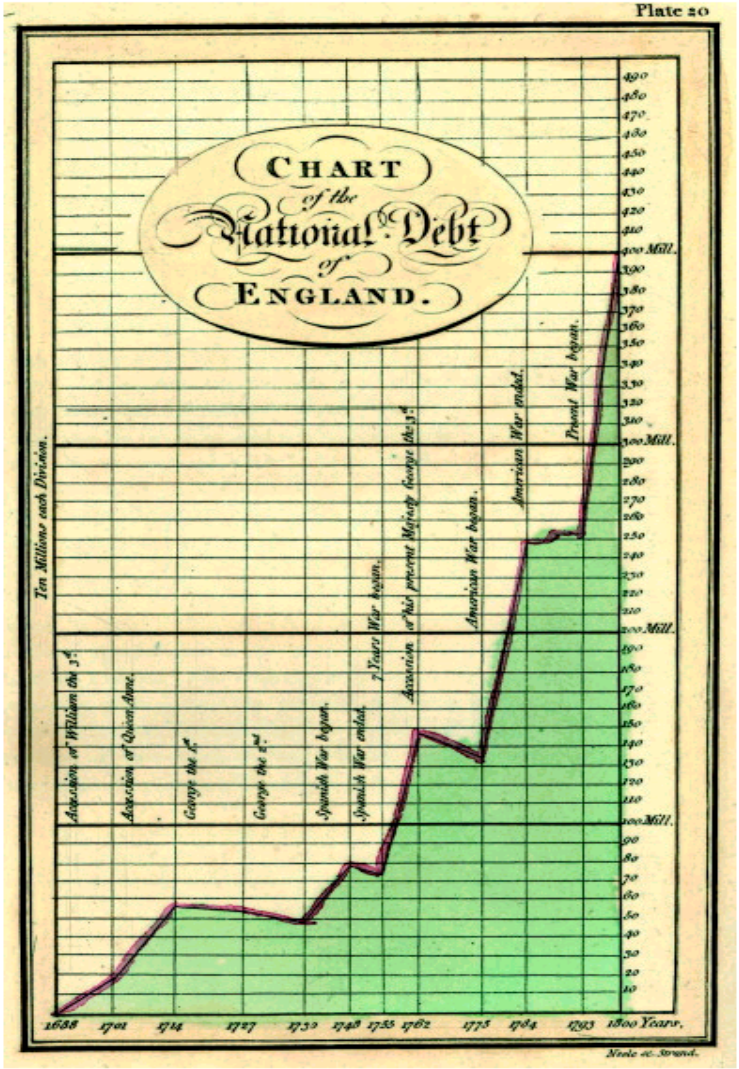
Pie chart, William Playfair

And to this



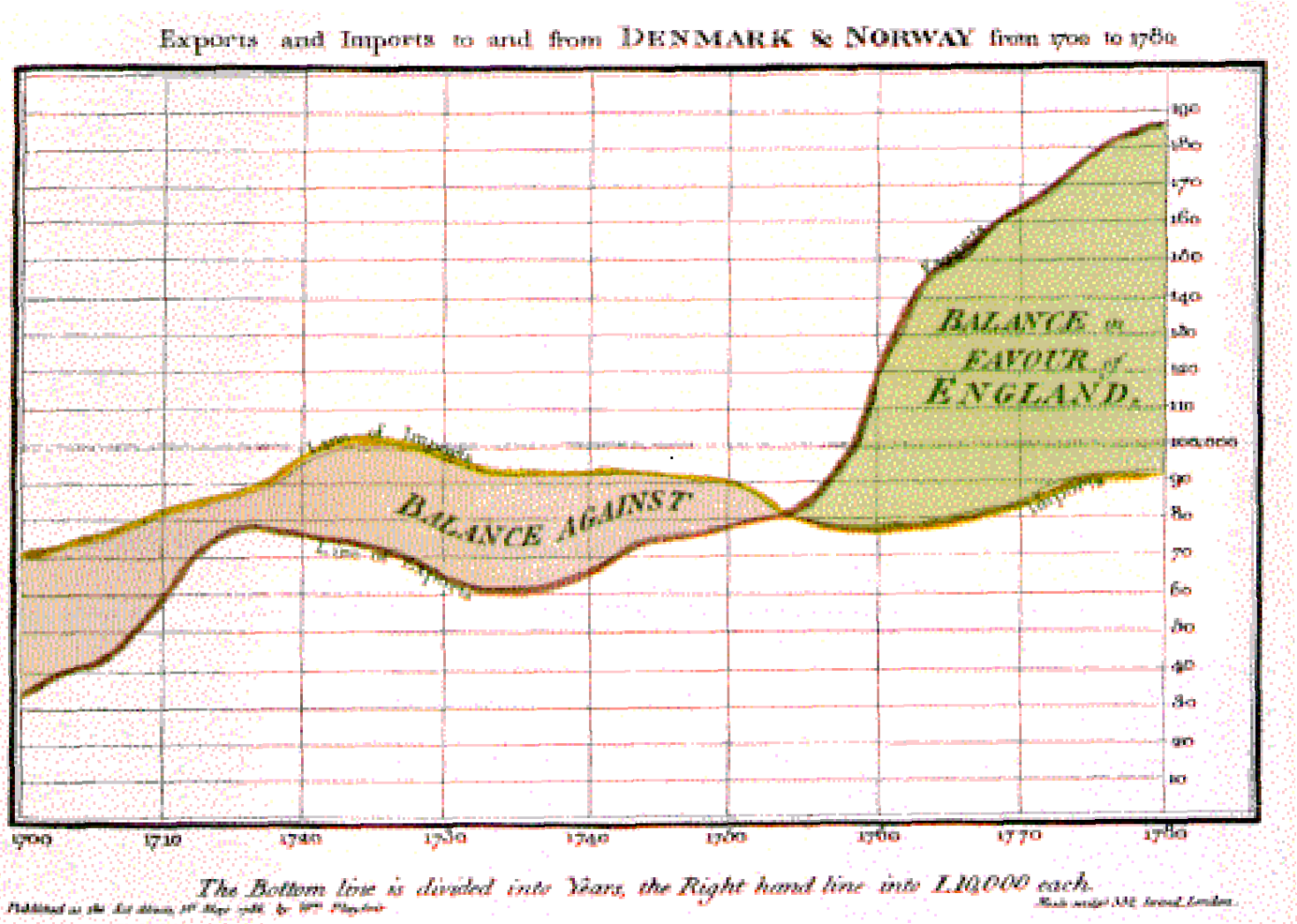
Composite line and bar chart, William Playfair

And this



Area chart, William Playfair

And this



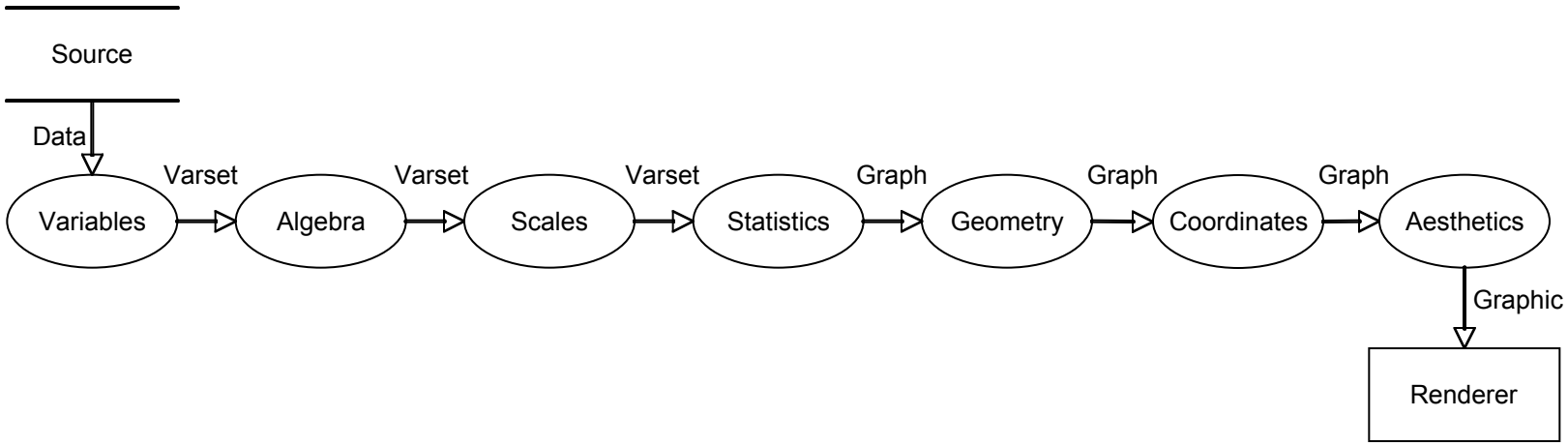
Differenced area chart, William Playfair

And this

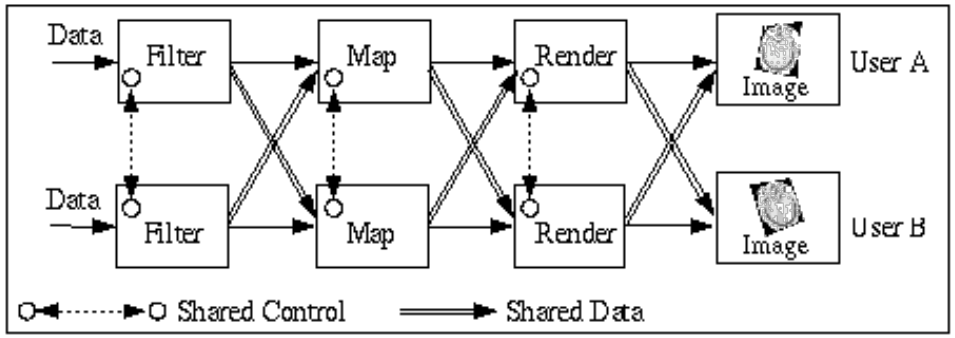
Soleto Map ca. 500 BCE, southern region of Italy's heel, discovered in dig supervised by Thierry van Compernolle, Montpellier University

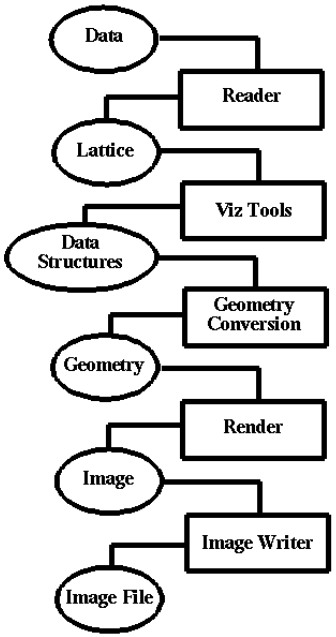
The GoG Function Chain

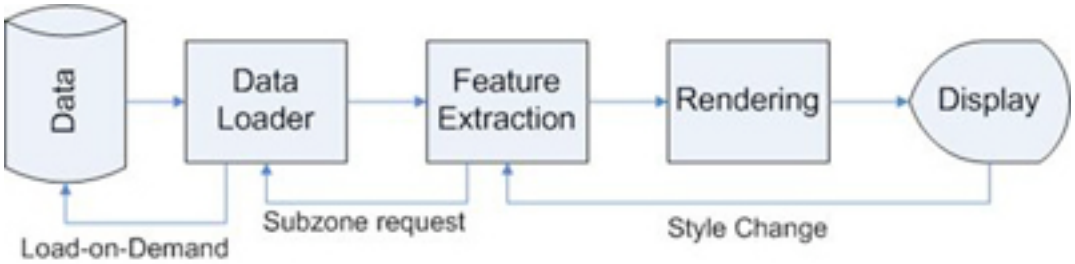
- Each ellipse is a class.
- Each class contains member functions.
  - functions are interchangeable within a class
- The chain is a total order.
  - changing this order produces a meaningless graphic
- The chain is invertible.



The GoG Chain is NOT a dataflow

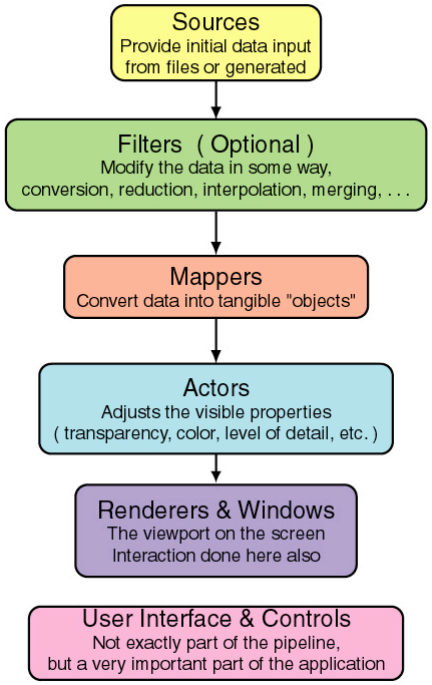


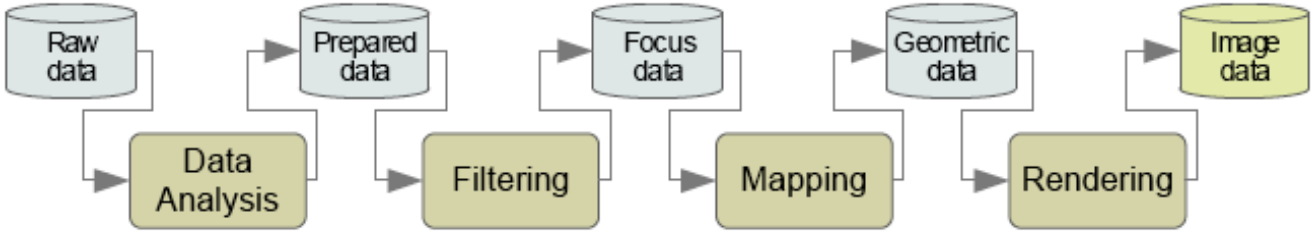


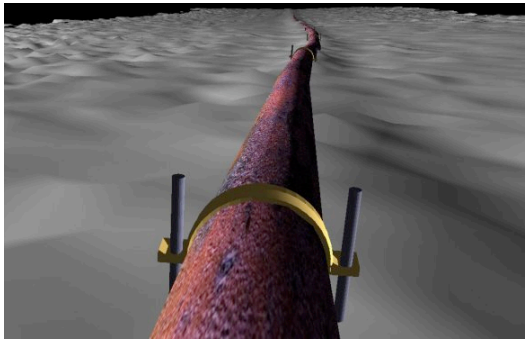


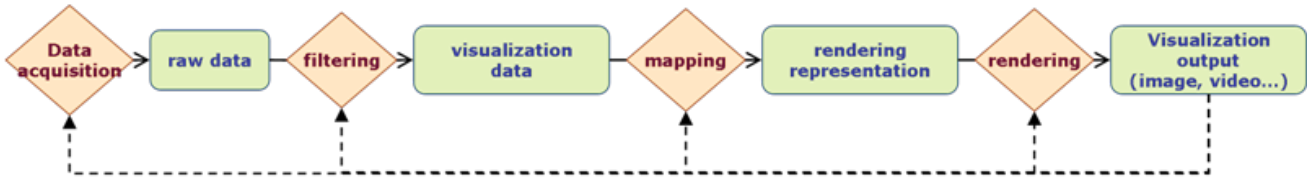
The GoG Chain is NOT a visualization pipeline

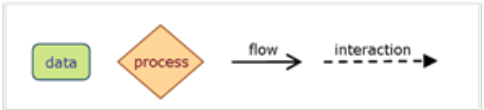
VTK Visualization Pipeline





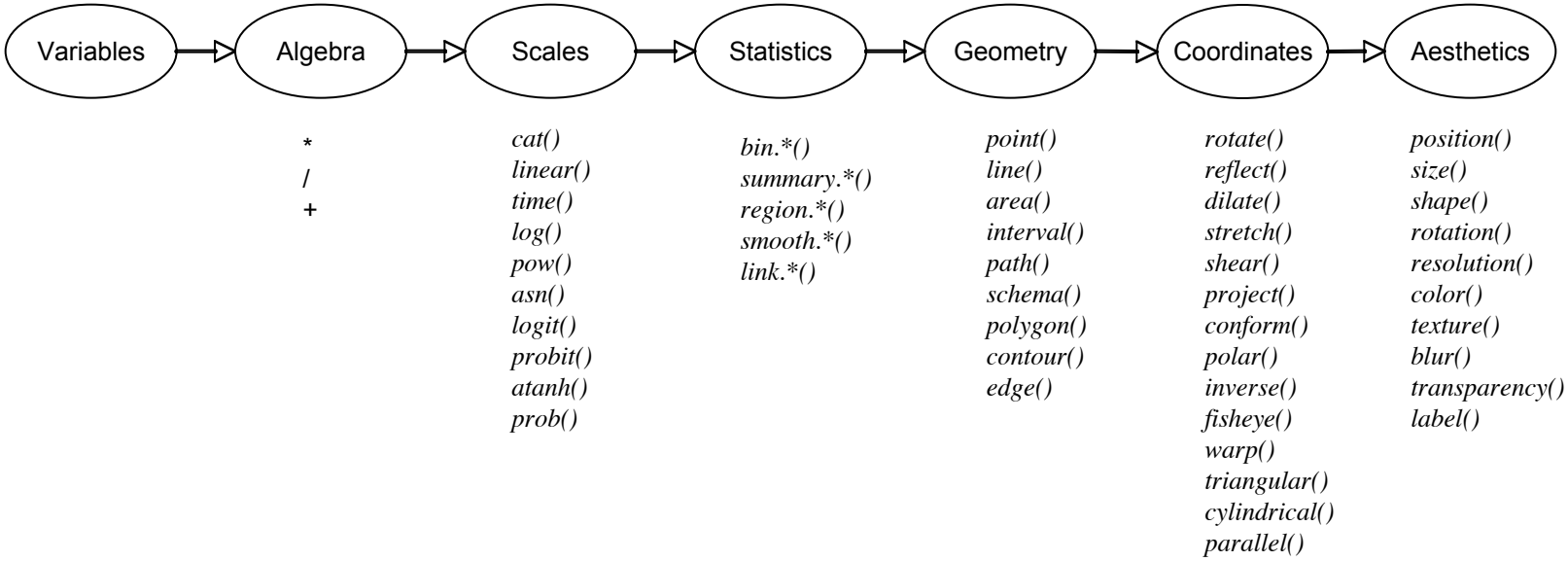




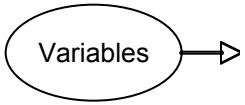


SYSTAT

GoG Member Functions



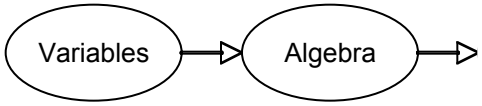
Variables



Variables

- Variables is a class that builds a data view.
- Variables receives a Dataset and outputs a Varset.
- A Dataset is a set of data.
- A Varset is a set of variables.
- A Variable is a function from a set of objects *O* to a set of values *V* (a many-to-one mapping).

Algebra



*
/
+

Algebra

- Algebra has three operators
  - cross (*)

x		a		x	a
y	*	a	=	y	a
z		b		z	b

  - nest (/)

x		a		x	a
y	/	a	=	y	a
z		b		z	b

  - blend (+)

x		a		x
y	+	a	=	y
z		b		z
				a
				a
				b

Algebra

Males	Impala
	Mustang
	Jaguar
Females	Impala
	Mustang
	Jaguar

animal * gender

- Male Jaguar and Female Jaguar are comparable

Algebra

Cars	Impala
	Mustang
	Jaguar
Animals	Impala
	Mustang
	Jaguar
Teams	Impala
	Mustang
	Jaguar

animal / class

- Jaguar Car and Jaguar Animal are not comparable

Algebra

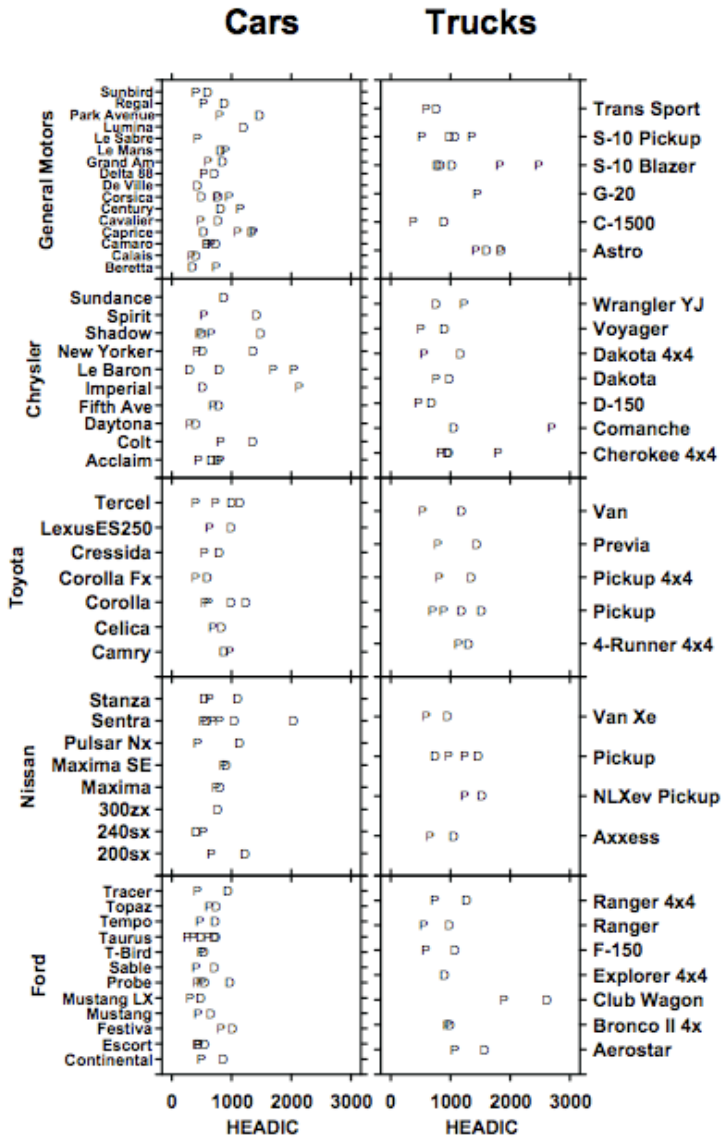
Cow
Chicken
Pig
Horse
Snake
Impala
Lion
Giraffe
Jaguar

barnyard + zoo

- Barnyard and Zoo must be from a common class or measurement scale for blend to be legal.

SYSTAT

Algebra


Cars
Trucks
General Motors
Sunbird
Regal
Park Avenue
Lumina
Le Sabre
Le Mans
Grand Am
Delta 88
De Ville
Corsica
Century
Cavalier
Caprice
Camaro
Calais
Beretta
Trans Sport
S-10 Pickup
S-10 Blazer
G-20
C-1500
Astro
Chrysler
Sundance
Spirit
Shadow
New Yorker
Le Baron
Imperial
Fifth Ave
Daytona
Colt
Acclaim
Wrangler YJ
Voyager
Dakota 4x4
Dakota
D-150
Comanche
Cherokee 4x4
Toyota
Tercel
LexusES250
Cressida
Corolla Fx
Corolla
Celica
Camry
Van
Previa
Pickup 4x4
Pickup
4-Runner 4x4
Nissan
Stanza
Sentra
Pulsar Nx
Maxima SE
Maxima
300zx
240sx
200sx
Van Xe
Pickup
NLXev Pickup
Axxess
Ford
Tracer
Topaz
Tempo
Taurus
T-Bird
Sable
Probe
Mustang LX
Mustang
Festiva
Escort
Continental
Ranger 4x4
Ranger
F-150
Explorer 4x4
Club Wagon
Bronco II 4x
Aerostar
0
1000
2000
3000
HEADIC

Algebra

The GoG algebra expression for this graphic is

H*T/(M*V)*O

The GoG algebra expression corresponds to the design model

$$H = C + M + V + O + T(MV) + MV + MO + VO + OT(MV) + MVO$$

where the symbols are

H: Head Injury Index

C: constant term (grand mean)

M: Manufacturer

V: Vehicle (car or truck)

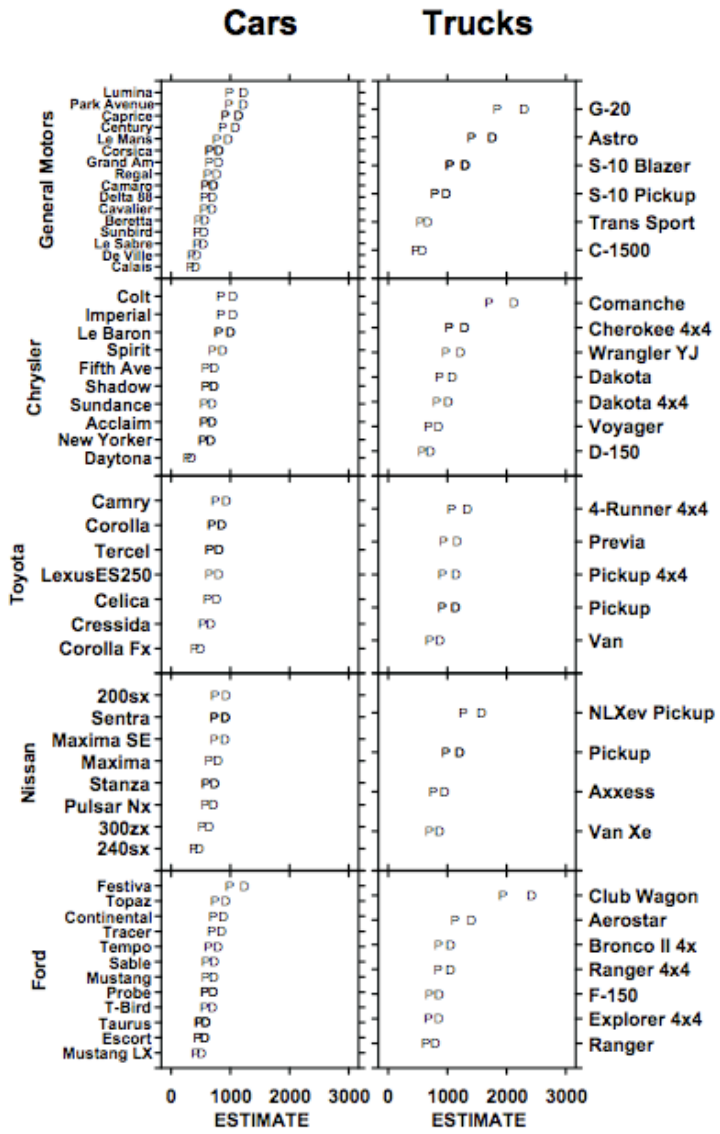
O: Occupant (driver or passenger)

T: Model

The smallest plausible subset model is:

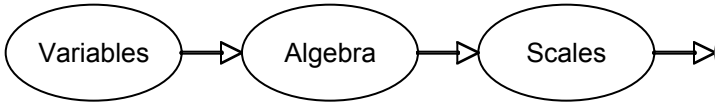
$$H = C + V + O + T(MV)$$

Algebra



SYSTAT

Scales



*
/
+

cat()
linear()
time()
log()
pow()
asn()
logit()
probit()
atanh()
prob()

Scales

- Scales map Varsets to Dimensions.
- A Dimension is a dimension of $\boldsymbol{R}^n$.
- Scales use one-to-one mappings within Frames
  - `identity(), log(), permutation(),...`
- More than one Varset can map to a Dimension.
- But we need rules for a meaningful mapping.
- Representational measurement is not sufficient.
  - nominal, ordinal, interval, ratio (Stevens)
  - too general for blend (+) operator
    - we can't blend area with weight
    - we can't blend speed with acceleration
    - we can't blend a density and a distribution function

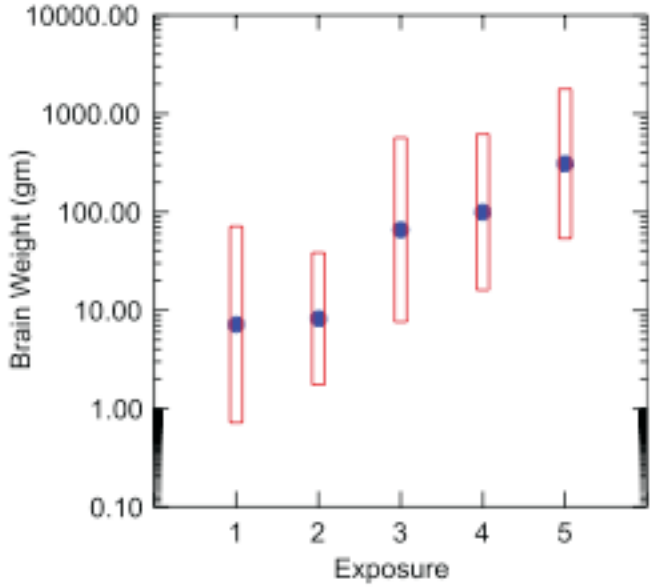
Scales

- We need measurement units.
  - more restrictive than axiomatic measurement scales

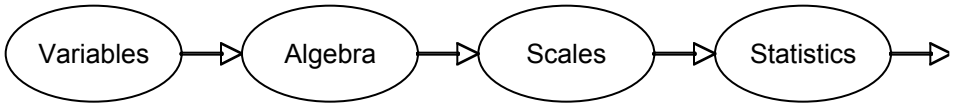
Length	*Mass*	*Time*	*Current*	*Temperature*
meter *point* *pica* *inch* *foot* *yard* *mile* *furlong* *fathom*	***kilogram*** *gram* *grain* *slug* *carat*	***second*** *minute* *hour* *day* *week* *month* *quarter* *year* *century*	***amp***	***kelvin*** *rankine* *celsius* *fahrenheit*
Substance	***Luminous Intensity***	***Angle***	***Count***	***Currency***
mole	***candela***	*radian* *degree* *minute* *second*	*unit* *dozen* *gross*	*dollar* *euro* *pound* *yen*

Scales

SCALE: *cat*(*dim*(1))
SCALE: *log*(*dim*(2), *base*(10))
ELEMENT: *point*(*position*(*summary.mean*(exposure*brainweight)))
ELEMENT: *interval*(*position*(*spread.sd*(exposure*brainweight),
color(*color.red*))



Statistics



*
/
+

cat()
linear()
time()
log()
pow()
asn()
logit()
probit()
atanh()
prob()

bin.()*
summary.()*
region.()*
smooth.()*
link.()*

Statistics

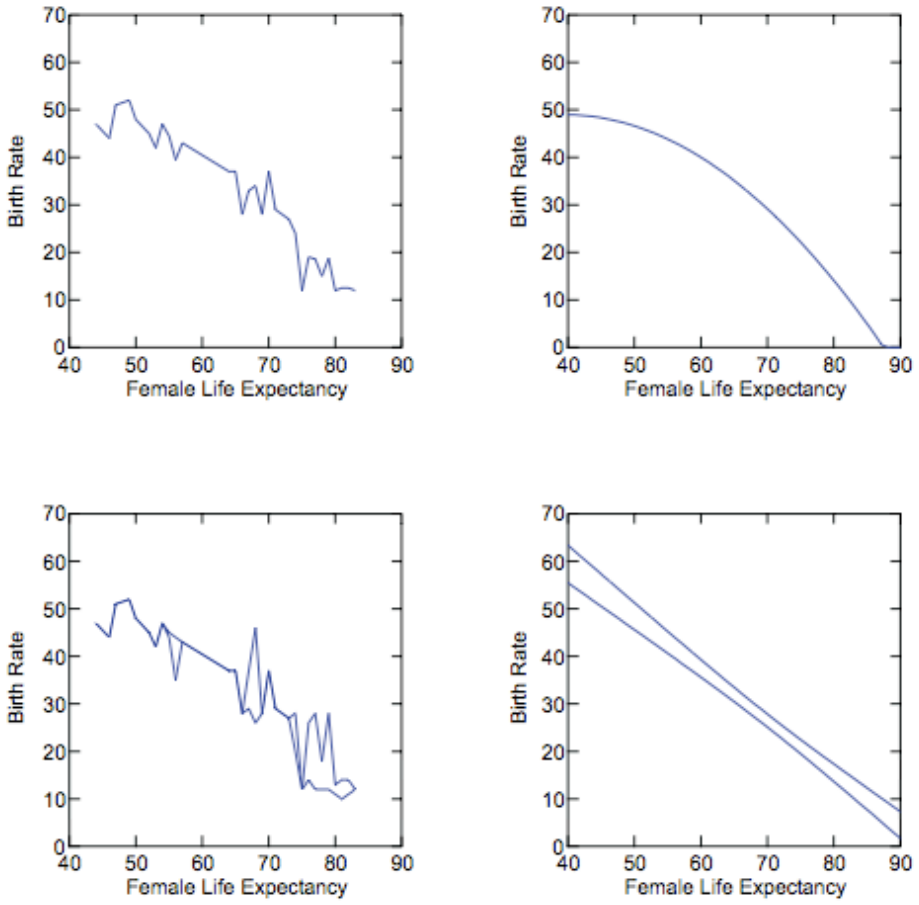
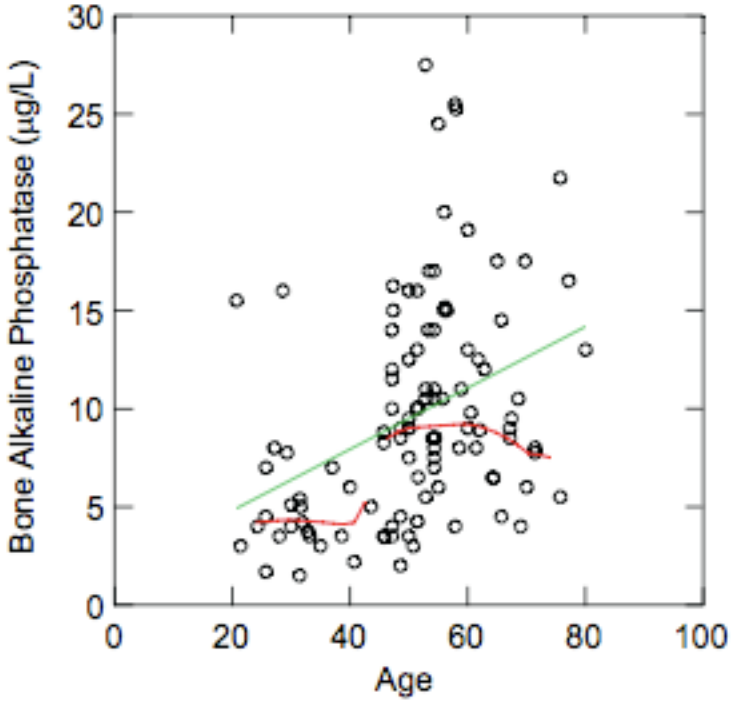


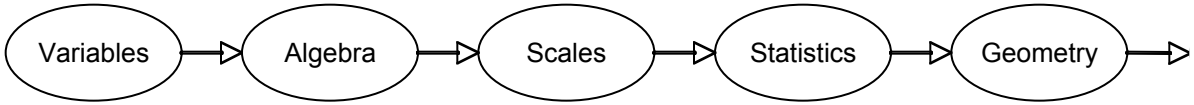
Figure 7.2 *Different statistical methods, same graph type*

Statistics

ELEMENT: *line*(*position*(*smooth.mode.epanechnikov*(age*bone)),
color(*color.red*))
ELEMENT: *line*(*position*(*smooth.linear*(age*bone)), *color*(*color.green*))
ELEMENT: *point*(*position*(age*bone))



Geometry



*
/
+

cat()
linear()
time()
log()
pow()
asn()
logit()
probit()
atanh()
prob()

bin.()*
summary.()*
region.()*
smooth.()*
link.()*

point()
line()
area()
interval()
path()
schema()
polygon()
contour()
edge()

Geometry

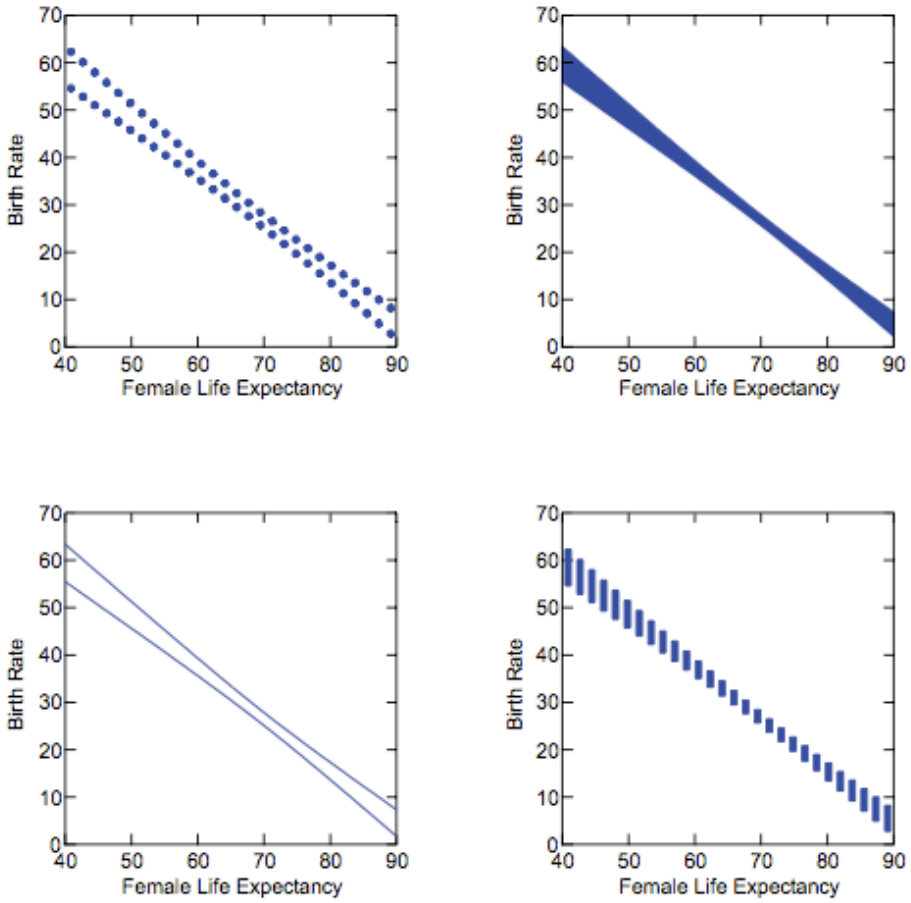


Figure 7.1 *Different graph types, same statistical method*

Statistics (recall)

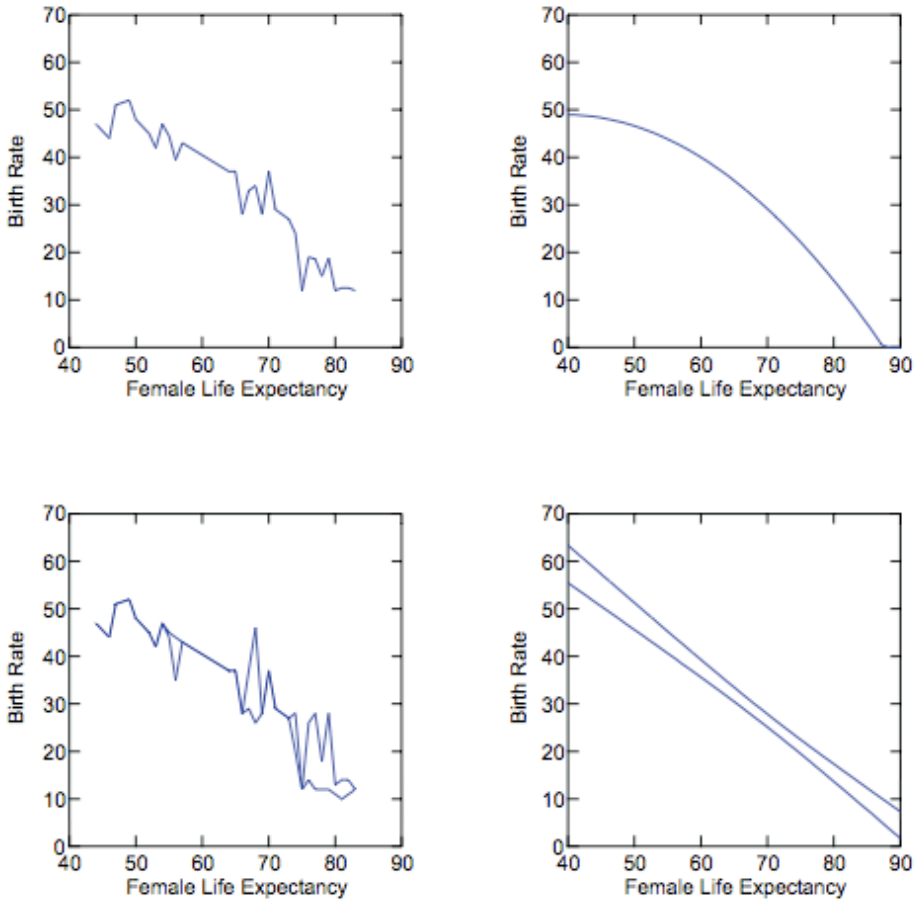
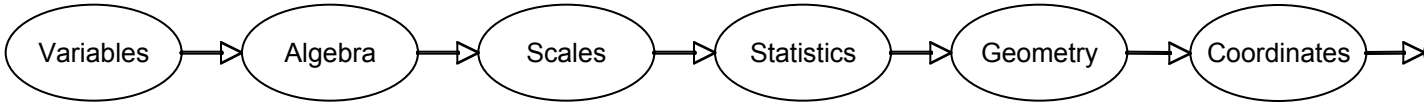


Figure 7.2 *Different statistical methods, same graph type*

Coordinates



*
/
+

cat()
linear()
time()
log()
pow()
asn()
logit()
probit()
atanh()
prob()

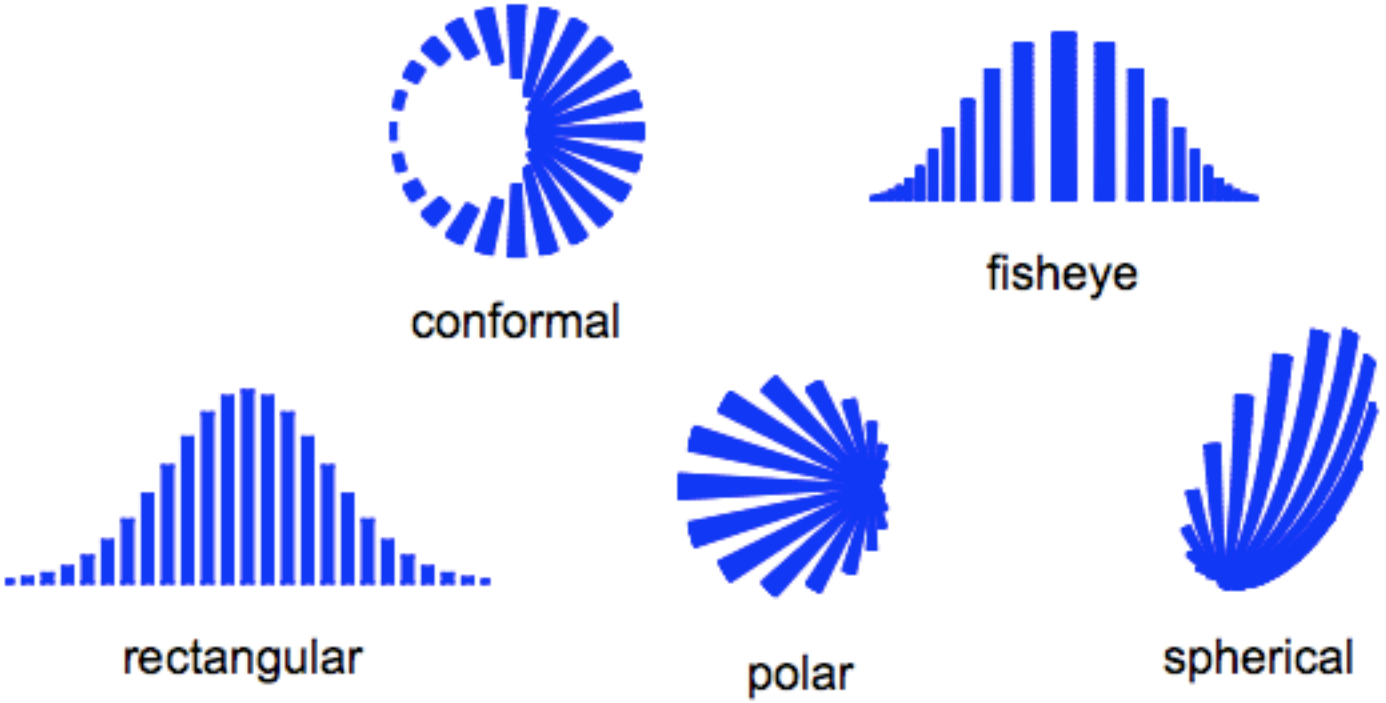
bin.()*
summary.()*
region.()*
smooth.()*
link.()*

point()
line()
area()
interval()
path()
schema()
polygon()
contour()
edge()

rotate()
reflect()
dilate()
stretch()
shear()
project()
conform()
polar()
inverse()
fisheye()
warp()
triangular()
cylindrical()
parallel()

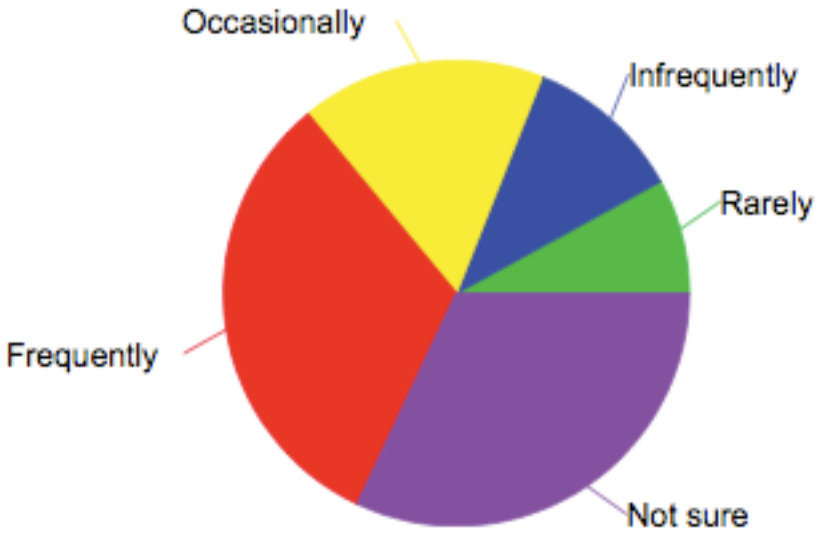
SYSTAT

Coordinates



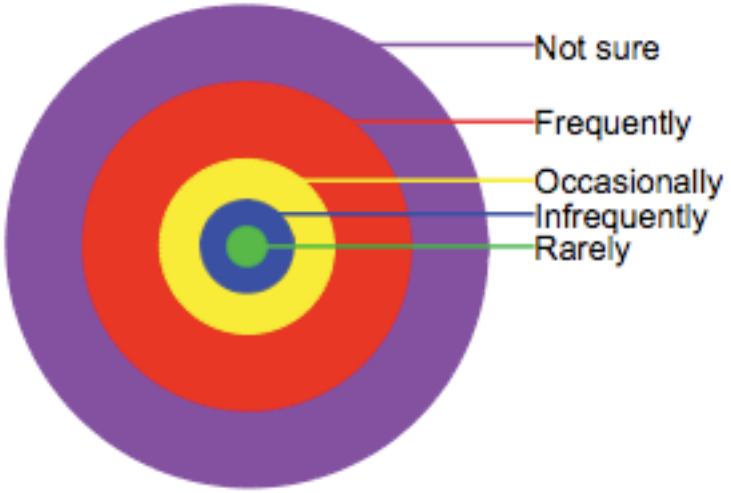
Coordinates

COORD: *polar.theta*()
ELEMENT: *interval.stack*(*position*(*summary.proportion*(response)),
label(response), *color*(response))



Coordinates

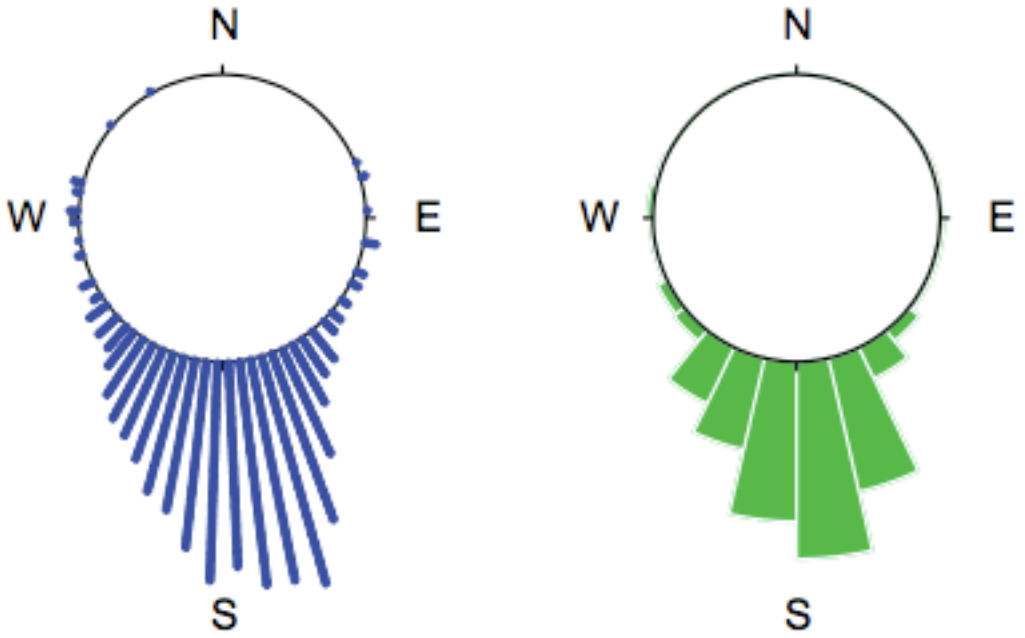
COORD: *polar.rho*()
ELEMENT: *interval.stack*(*position*(*summary.proportion*(response)),
label(response), *color*(response))



Coordinates

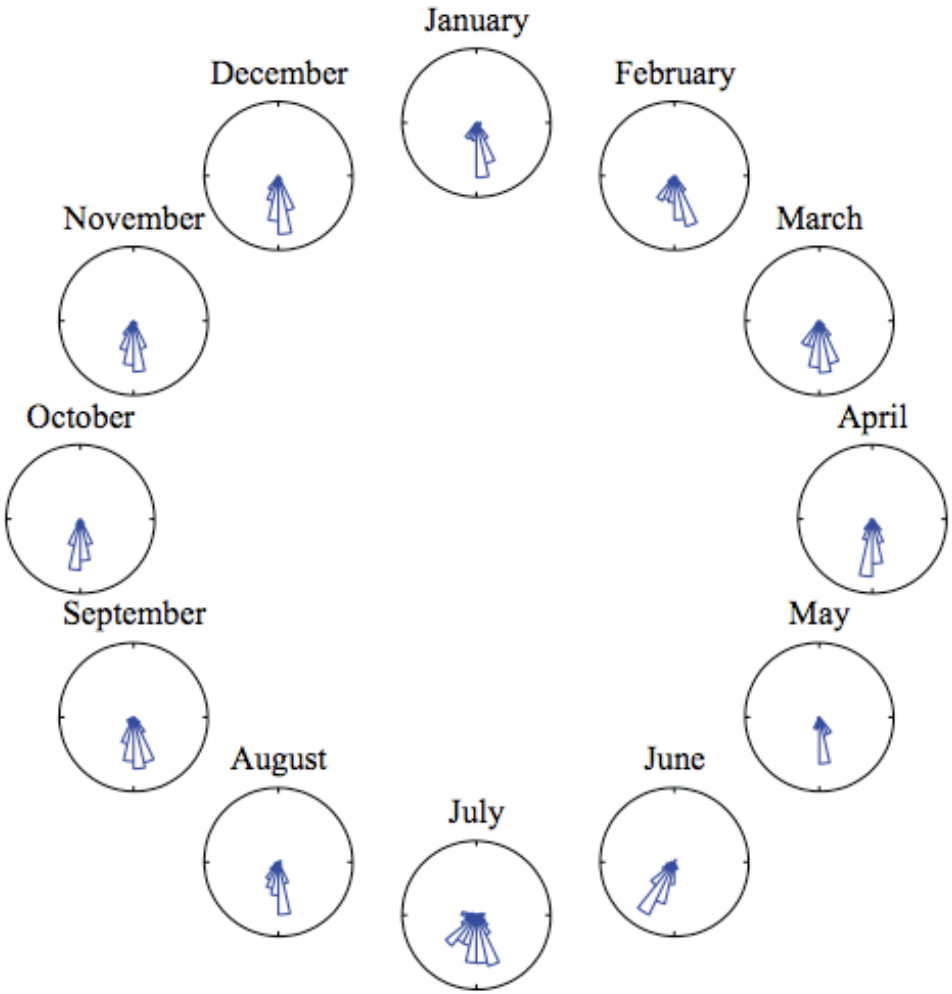
COORD: *polar.rho.plus*()
ELEMENT: *point.dodge*(*position*(*bin.dot*(direction)))

COORD: *polar.rho.plus*()
ELEMENT: *interval*(*position*(*summary.count*(*bin.rect*(direction))))

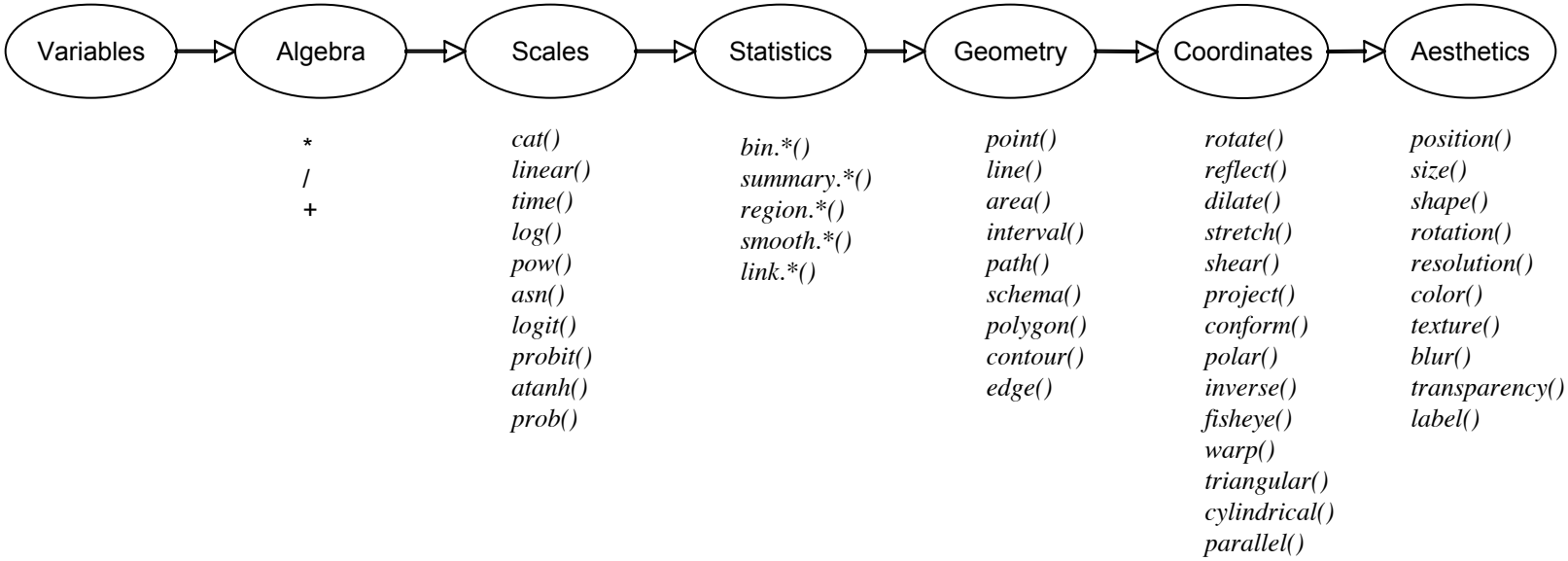


Coordinates

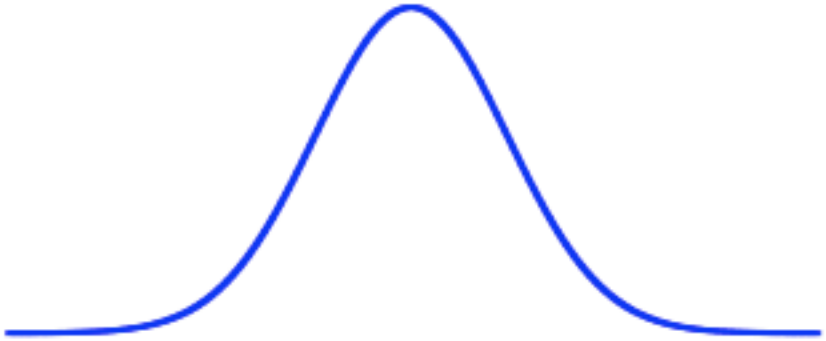
COORD: *polar.theta*(*dim*(3), *polar*(*dim*(1, 2)))
ELEMENT: *interval*(*position*(*summary.count*(*bin.rect*(month*direction, *dim*(2)))))



Aesthetics


Variables
Algebra
Scales
Statistics
Geometry
Coordinates
Aesthetics
*
/
+
cat()
linear()
time()
log()
pow()
asn()
logit()
probit()
atanh()
prob()
bin.*()
summary.*()
region.*()
smooth.*()
link.*()
point()
line()
area()
interval()
path()
schema()
polygon()
contour()
edge()
rotate()
reflect()
dilate()
stretch()
shear()
project()
conform()
polar()
inverse()
fisheye()
warp()
triangular()
cylindrical()
parallel()
position()
size()
shape()
rotation()
resolution()
color()
texture()
blur()
transparency()
label()

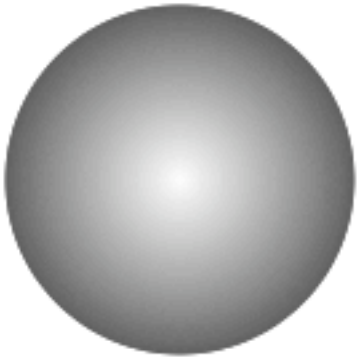
Aesthetics



position

blur

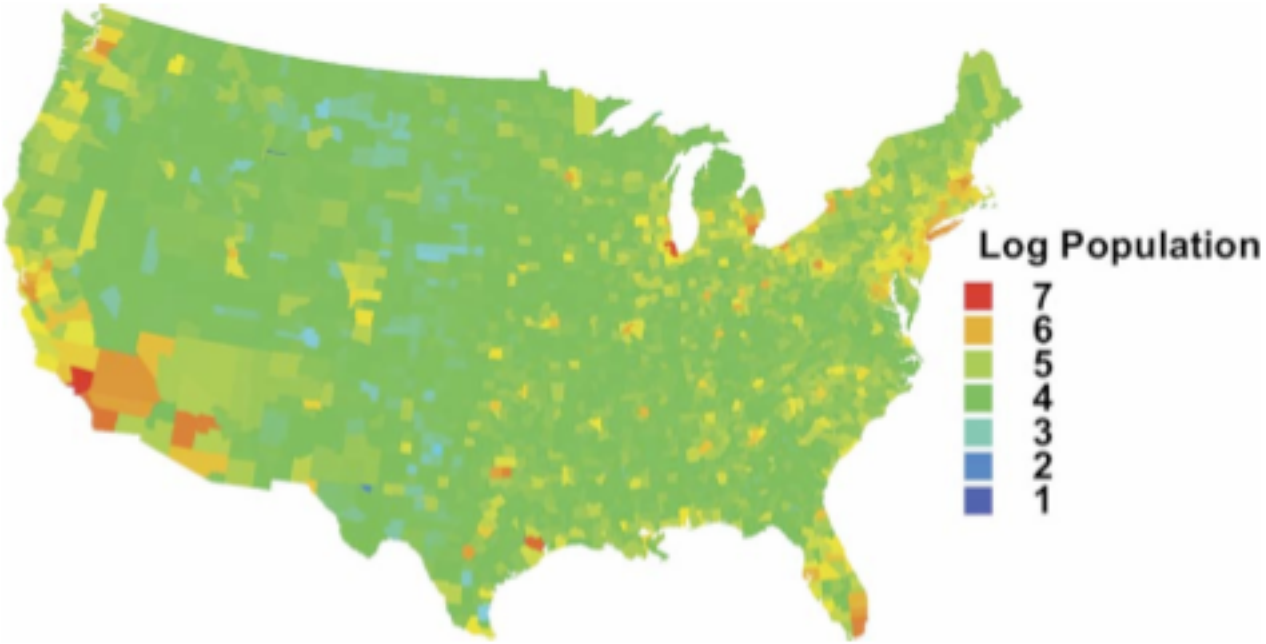
sound



brightness

Aesthetics

DATA: longitude, latitude = *map*(*source*("US counties"))
TRANS: lpop = *log.10*(population)
COORD: *project.stereo*(*dim*(1, 2))
ELEMENT: *polygon*(*position*(longitude*latitude), *color.hue*(lpop))



Four GoG Implementations

SPSS nViZn — Wilkinson, Wills, Rope, ...

Stolte, Hanrahan, Mackinlay, ...

Wickham

Protovis — Bostock, Heer

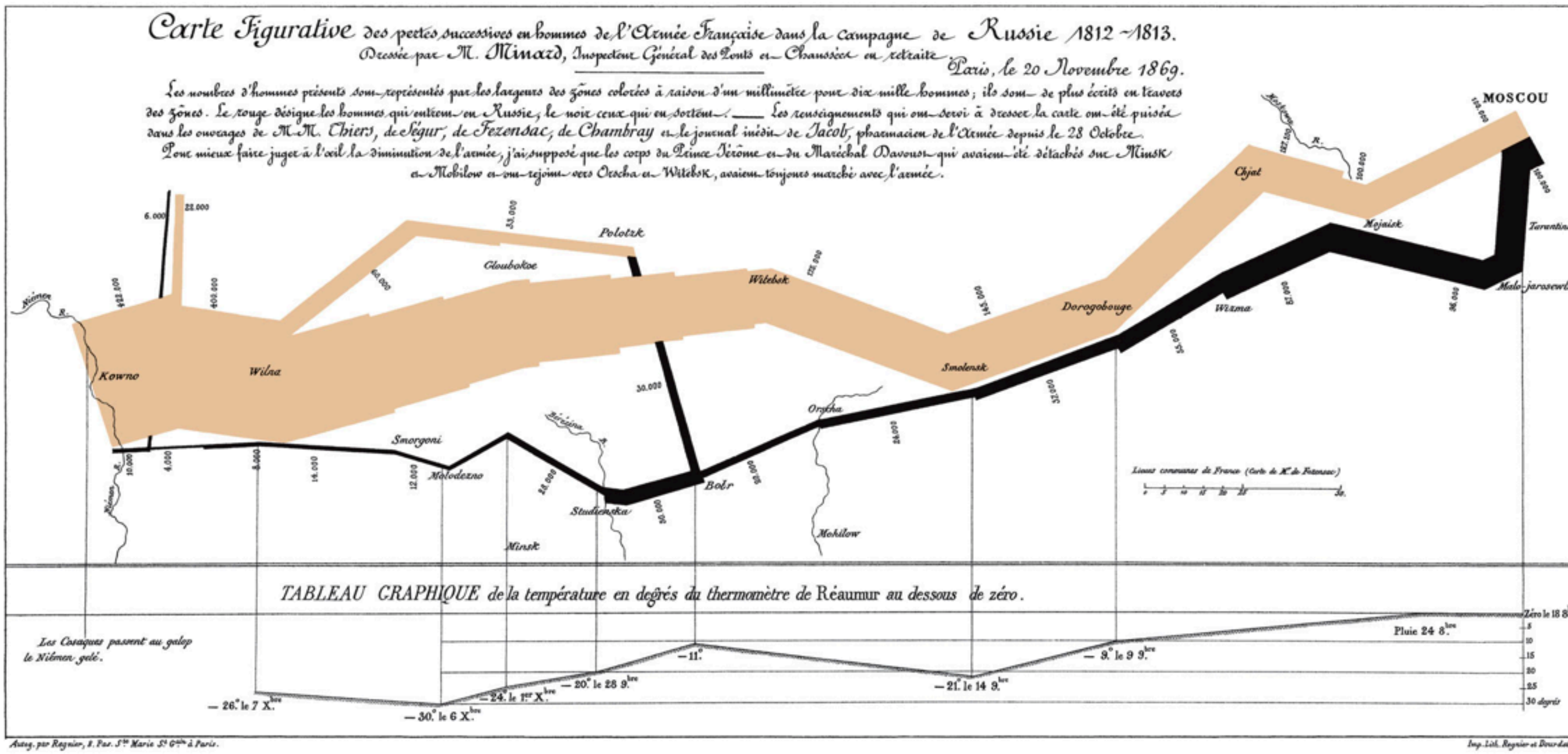
Aspects of a Formal Graphics Language

- These are adjectives we sometimes use to describe formal languages:
  - Simple - few primitives, operators, and functions
  - Expressive - produces a large result set
  - Coherent - based on a grammar
  - Meaningful - makes statements that are falsifiable

Simple

- Simplicity in a language means having few primitives, operators, and functions.
- Let’s take a look at a famous example and see how simple it is to build it in various graphics languages...

Simple

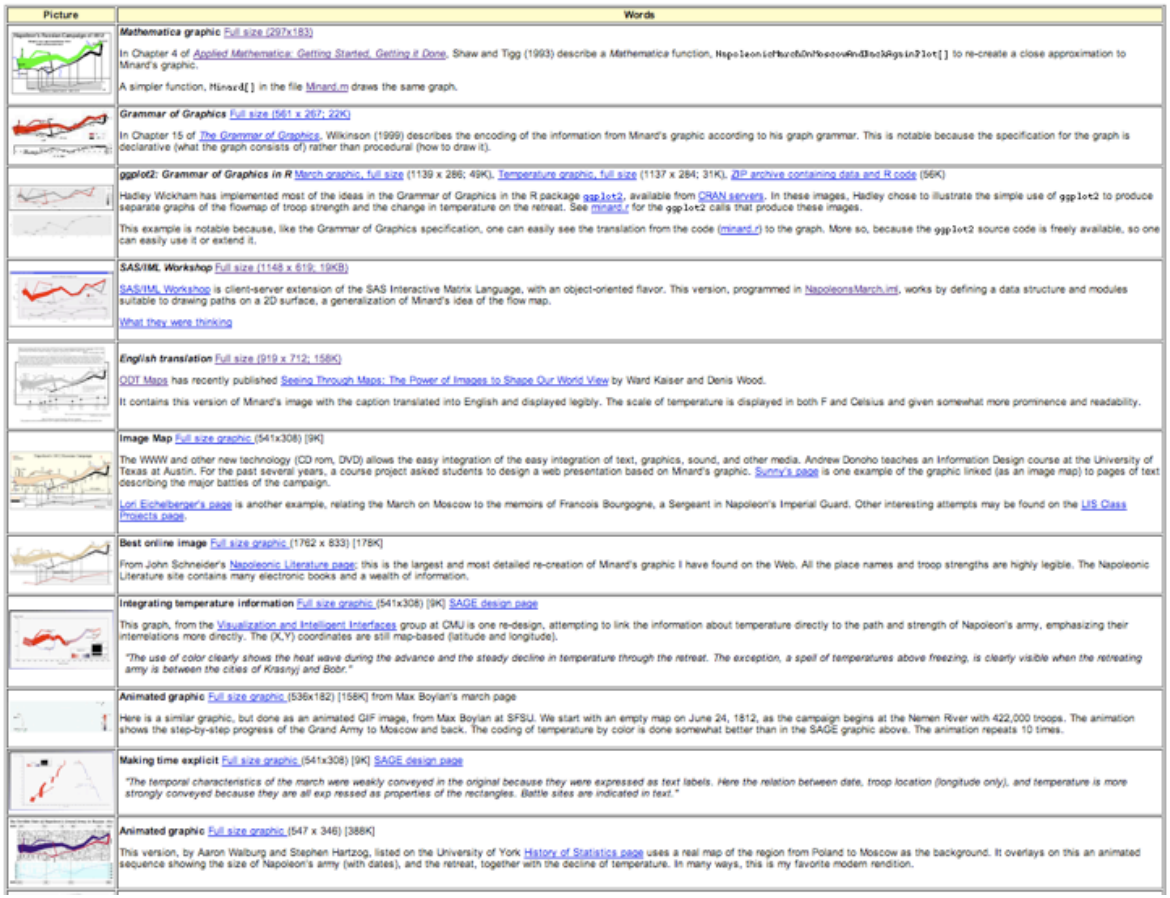


Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812–1813.
Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite.
Paris, le 20 Novembre 1869.
Les nombres d'hommes présents sont représentés par les largeurs des zones colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zones. Le rouge désigne les hommes qui entrent en Russie, le noir ceux qui en sortent. —— Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M.M. Thiers, de Ségur, de Fezensac, 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 Davoust qui avaient été détachés sur Minsk et Mobilow et ont rejoint vers Orscha et Witebsk, avaient toujours marché avec l'armée.
MOSCOU
Kowno
Wilna
Gloubokoe
Polotzk
Witebsk
Smolensk
Dorogobouge
Chjat
Mojaisk
Wizma
Tarantino
Malo-jarosewli
Orscha
Bobr
Studienska
Smorgoni
Molodezno
Minsk
Mohilow
TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.
Les Cosaques passent au galop le Niémen gelé.
Zéro le 18 8bre
Pluie 24 8bre
— 9° le 9 9bre
— 21° le 14 9bre
— 11°
— 20° le 28 9bre
— 24° le 1er Xbre
— 30° le 6 Xbre
— 26° le 7 Xbre

Simple

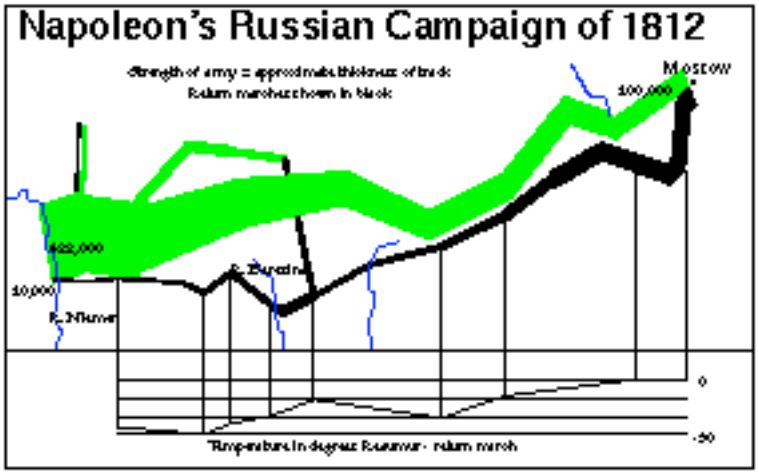
You can find the following examples at this website:

http://www.math.yorku.ca/SCS/Gallery/re-minard.html



Picture	Words
	Mathematica graphic Full size (297x183) In Chapter 4 of *Applied Mathematica: Getting Started, Getting it Done*, Shaw and Tigg (1993) describe a Mathematica function, `NapoleonicMarch[]` to re-create a close approximation to Minard's graphic. A simpler function, `Minard[]` in the file Minard.m draws the same graph.
	Grammar of Graphics Full size (561 x 267; 22K) In Chapter 15 of *The Grammar of Graphics*, Wilkinson (1999) describes the encoding of the information from Minard's graphic according to his graph grammar. This is notable because the specification for the graph is declarative (what the graph consists of) rather than procedural (how to draw it).
	ggplot2: Grammar of Graphics in R March graphic, full size (1139 x 286; 49K), Temperature graphic, full size (1137 x 284; 31K), ZIP archive containing data and R code (56K) Hadley Wickham has implemented most of the ideas in the Grammar of Graphics in the R package `ggplot2`, available from CRAN servers. In these images, Hadley chose to illustrate the simple use of `ggplot2` to produce separate graphs of the flowmap of troop strength and the change in temperature on the retreat. See minard.r for the `ggplot2` calls that produce these images. This example is notable because, like the Grammar of Graphics specification, one can easily see the translation from the code (minard.r) to the graph. More so, because the `ggplot2` source code is freely available, so one can easily use it or extend it.
	SAS/IML Workshop Full size (1148 x 619; 19KB) SAS/IML Workshop is client-server extension of the SAS Interactive Matrix Language, with an object-oriented flavor. This version, programmed in NapoleonsMarch.iml, works by defining a data structure and modules suitable to drawing paths on a 2D surface, a generalization of Minard's idea of the flow map. What they were thinking
	English translation Full size (919 x 712; 158K) ODT Maps has recently published Seeing Through Maps: The Power of Images to Shape Our World View by Ward Kaiser and Denis Wood. It contains this version of Minard's image with the caption translated into English and displayed legibly. The scale of temperature is displayed in both F and Celsius and given somewhat more prominence and readability.
	Image Map Full size graphic (541x308) [9K] The WWW and other new technology (CD rom, DVD) allows the easy integration of the easy integration of text, graphics, sound, and other media. Andrew Donoho teaches an Information Design course at the University of Texas at Austin. For the past several years, a course project asked students to design a web presentation based on Minard's graphic. Sunny's page is one example of the graphic linked (as an image map) to pages of text describing the major battles of the campaign. Lori Eichelberger's page is another example, relating the March on Moscow to the memoirs of Francois Bourgogne, a Sergeant in Napoleon's Imperial Guard. Other interesting attempts may be found on the UIS Class Projects page.
	Best online image Full size graphic (1762 x 833) [178K] From John Schneider's Napoleonic Literature page: this is the largest and most detailed re-creation of Minard's graphic I have found on the Web. All the place names and troop strengths are highly legible. The Napoleonic Literature site contains many electronic books and a wealth of information.
	Integrating temperature information Full size graphic (541x308) [9K] SAGE design page This graph, from the Visualization and Intelligent Interfaces group at CMU is one re-design, attempting to link the information about temperature directly to the path and strength of Napoleon's army, emphasizing their interrelations more directly. The (X,Y) coordinates are still map-based (latitude and longitude). *"The use of color clearly shows the heat wave during the advance and the steady decline in temperature through the retreat. The exception, a spell of temperatures above freezing, is clearly visible when the retreating army is between the cities of Krasnyj and Bobr."*
	Animated graphic Full size graphic (536x182) [158K] from Max Boylan's march page Here is a similar graphic, but done as an animated GIF image, from Max Boylan at SFSU. We start with an empty map on June 24, 1812, as the campaign begins at the Nemen River with 422,000 troops. The animation shows the step-by-step progress of the Grand Army to Moscow and back. The coding of temperature by color is done somewhat better than in the SAGE graphic above. The animation repeats 10 times.
	Making time explicit Full size graphic (541x308) [9K] SAGE design page *"The temporal characteristics of the march were weakly conveyed in the original because they were expressed as text labels. Here the relation between date, troop location (longitude only), and temperature is more strongly conveyed because they are all exp ressed as properties of the rectangles. Battle sites are indicated in text."*
	Animated graphic Full size graphic (547 x 346) [388K] This version, by Aaron Walburg and Stephen Hartzog, listed on the University of York History of Statistics page uses a real map of the region from Poland to Moscow as the background. It overlays on this an animated sequence showing the size of Napoleon's army (with dates), and the retreat, together with the decline of temperature. In many ways, this is my favorite modern rendition.

Simple? - Mathematica



Simple? - Mathematica

```
BeginPackage["Minard`"]

Minard::usage = "Minard.m is a package for producing thematic maps
in the style of C. J. Minard"

ProcessTemp::usage = "ProcessTemp[list] computes the graphics primitives to be associated with a list of elements
of the form {x, y, temp}"

stuff::usage = "stuff[list] computes the graphics primitives to be associated with a list of elements of the form
{x, y, strength}"

ProcessStrength::usage = "ProcessStrength[list] computes the graphics primitives to be associated with a list of
elements of the form {{x1, y1, strength1},....., {xn, yn, strengthn}}"

ProcessRivers::usage = "ProcessRivers[list] computes the graphics primitives to be associated with a list of river
coordinates"

ProcessPoints::usage = "ProcessPoints[list] computes the graphics primitives to be associated with a list of city
or other point coordinates"

ProcessTitle::usage = "ProcessTitle[location, title] computes the graphics primitives to be associated with the
title"

ProcessBoxes::usage = "ProcessBoxes[list] computes the graphics primitives to be associated with a list of box
coordinates"

ProcessText::usage = "ProcessText[list] computes the graphics primitives to be associated with a list of
coordinates and associated text"

NapoleonicMarchOnMoscowAndBackAgainPlot::usage =
"NapoleonicMarchOnMoscowAndBackAgainPlot[strength_list, temp_list, riverdata_list, boxdata_list, titledata_list,
pointdata_list, textdata_list] shows the thematic chart associated with the strength temperature, river, box,
title, point and text lists, in the style of C. J. Minard."
```

Simple? - Mathematica

```
Begin["`Private`"]
ProcessTemp[tdata_] :=
Module[{tprimlist, coords = Array[coordarr, 100], numb, k,
temprules, firstx, lastx},
temprules = {0, -10, -20, -30};
firstx = First[tdata][[1]];
lastx = Last[tdata][[1]];
tprimlist = Map[({Thickness[0.001], Line[{{firstx, 0.0916 + 0.00188333*#},
{lastx, 0.0916 + 0.00188333*#}}]})&, temprules];
numb = Length[tdata];
Do[
coords[[k]] = {tdata[[k, 1]], 0.0916 + tdata[[k,3]]*0.0018833}
,{k, 1, numb}];
Do[
tprimlist = Append[tprimlist,
{Thickness[0.001], Line[{tdata[[k, Range[2]]] , coords[[k]]} ]}];
,{k, 1, numb}];
Do[
tprimlist = Append[tprimlist,
{Thickness[0.001], Line[{coords[[k]] , coords[[k+1]]} ]}];
,{k, 1, numb-1}];
tprimlist]

stuff[sd_] := Module[{k, m, primlist,pca, pcb, pcc, pcd,
scale, dir = Array[dirarr, 100],
rotdir = Array[rotdirarr, 100],
l = Array[larr, 100],
avevec  = Array[avevecarr, 100]},
scale = 10500000;
primlist = { };
k = Length[sd];
```

Simple? - Mathematica

```
Do[(dir[[m]] = sd[[m+1, Range[2] ]] - sd[[m, Range[2] ]];
l[[m]] = Sqrt[dir[[m, 1]]^2 + dir[[m, 2]]^2]),
{m, 1, k-1}];
Do[
If[l[[m]] > 0.0001,
rotdir[[m]] = {-dir[[m, 2]], dir[[m, 1]]}/l[[m]],
rotdir[[m]] =
{-dir[[m+1, 2]], dir[[m+1, 1]]}/l[[m+1]]],
{m, 1, k-1}];
avevec[[1]] = rotdir[[1]];
avevec[[k]] = rotdir[[k-1]];
Do[
avevec[[m]] =
(rotdir[[m]] + rotdir[[m-1]])/(1 +
rotdir[[m,1]]*rotdir[[m-1,1]]
+ rotdir[[m,2]]*rotdir[[m-1,2]]), {m, 2, k-1}];
Do[
pca = sd[[m,Range[2]]] - sd[[m,3]]*avevec[[m]]/scale;
pcb = sd[[m+1,Range[2]]] - sd[[m+1,3]]*avevec[[m+1]]/scale;
pcc = sd[[m+1,Range[2]]] + sd[[m+1,3]]*avevec[[m+1]]/scale;
pcd = sd[[m,Range[2]]] + sd[[m,3]]*avevec[[m]]/scale;
u = {If[sd[[m, 4]] == 1,
RGBColor[0,1,0],
RGBColor[0,0,0]],
Polygon[{pca, pcb, pcc, pcd}]};
primlist = Append[primlist, u],
{m, 1, k-1}];
primlist]
```

Simple? - Mathematica

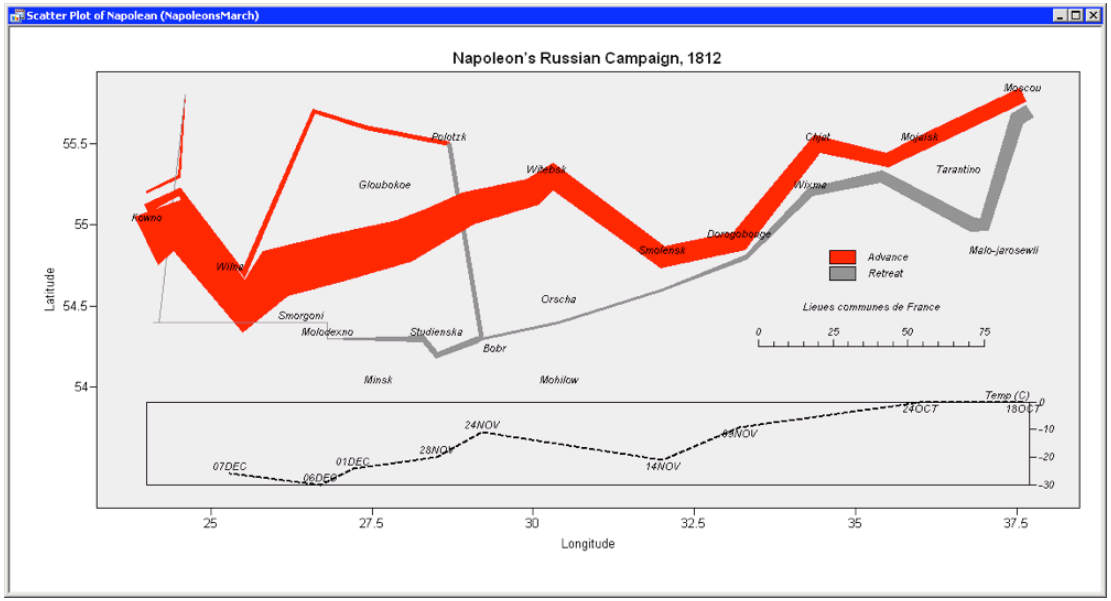
```
ProcessStrength[data_] :=
Module[{strengthprims, str, len, v},
strengthprims = {};
len = Length[data];
Do[
(strengthprims =
Append[strengthprims, stuff[data[[v]]]]),
{v, 1, len}];
strengthprims
]

ProcessTitle[titledata_] :=
Text[FontForm[titledata[[1, 3]],{"Helvetica-Bold", 14}],
 {titledata[[1, 1]], titledata[[1, 2]]}]

ProcessPoints[pointdata_] :=
Map[({Point[Drop[#, -1]], Text[FontForm[Last[#], {"Times-Roman",   8}],
Drop[#, -1]+{0.01,0.01}, {0, -1}]})&, pointdata]

ProcessText[textdata_] :=
Map[(Text[FontForm[Last[#], {"Times-Roman",   6}], Drop[#, -1], {0, -1}])&,  textdata]

ProcessRivers[riverdata_] :=
Map[({RGBColor[0, 0, 1], Thickness[0.001], Line[#]}&), riverdata]

ProcessBoxes[boxdata_] :=
Map[({RGBColor[0, 0, 0], Thickness[0.002], Line[#]}&), boxdata]
```

SYSTAT

Simple? - Mathematica

```
NapoleonicMarchOnMoscowAndBackAgainPlot[sdata_, tdata_, riverdata_, boxdata_, titledata_, pointdata_, textdata_] :=
Show[Graphics[
{ProcessStrength[sdata],
 ProcessTemp[tdata],
 ProcessRivers[riverdata],
 ProcessBoxes[boxdata],
 ProcessTitle[titledata],
 ProcessPoints[pointdata],
 ProcessText[textdata]}
]]

Minard[] :=
NapoleonicMarchOnMoscowAndBackAgainPlot[
     StrengthData,
     TempData,
     RiverData,(* river *)
     BoxData,
     TitleData,(* title *)
     PointData,(* point *)
     TextData       (* text *)
     ]

TempData = {...};

RiverData = {...};

armydata = {...};

StrengthData = {...};

BoxData = {...};

TitleData = {...};
PointData = {...};
TextData = {...};
End[]
EndPackage[]
```

Simple? - SAS


Scatter Plot of Napolean (NapoleonsMarch)
Napoleon's Russian Campaign, 1812
Latitude
Longitude
55.5
55
54.5
54
25
27.5
30
32.5
35
37.5
Kowno
Wilna
Smorgoni
Molodexno
Minsk
Studienska
Bobr
Mohilow
Orscha
Polotzk
Gloubokoe
Witebsk
Smolensk
Dorogobouge
Wixma
Chiat
Moiaisk
Moscou
Tarantino
Malo-jarosewli
Advance
Retreat
Lieues communes de France
0
25
50
75
Temp (C)
0
-10
-20
-30
18OCT
24OCT
28OCT
04NOV
14NOV
24NOV
28NOV
01DEC
06DEC
07DEC

Simple? - SAS

```
run DefineData;

declare DataObject dobj;
dobj = DataObject.Create( "Napolean", {"Longitude" "Latitude" "ArmySize" "ArmyDir" "ArmyGroup"}, army );

declare ScatterPlot plot;
plot = ScatterPlot.Create( dobj, "Longitude", "Latitude", false );
plot.SetWindowPosition( 0, 25, 100, 75 );
plot.SetGraphAreaMargins( 0.08, 0.02, 0.08, 0.15 );
plot.SetAxisViewRange( YAXIS, cityLat[><], cityLat[<>] );
plot.ShowObs( false );
plot.SetTitleText( "Napoleon's Russian Campaign, 1812" );
plot.ShowTitle();
plot.DrawUseDataCoordinates();

sizeScale = 6000;
group = unique(armyGroup);
do i = ncol(group) to 1 by -1;
   idx = loc( armyGroup = group[i] );
   /* group's advance */
   forward = loc( armyDir[idx] = 1 );
   plot.DrawSetPenColor( RED );
   plot.DrawSetBrushColor( RED );
   k = idx[forward];
   run PlotContinuousDataPath( armyLon[k], armyLat[k], armySize[k]/sizeScale, plot );

   /* group's retreat */
   backward = loc( armyDir[idx] = -1 );
   plot.DrawSetPenColor( GREY );
   plot.DrawSetBrushColor( GREY );
   k = idx[backward];
   run PlotContinuousDataPath( armyLon[k], armyLat[k], armySize[k]/sizeScale, plot );
end;
```

Simple? - SAS

```
/* label cities for geographical reference */
plot.DrawSetTextSize( 8 );
plot.DrawSetTextStyle( STYLE_ITALIC );
plot.DrawSetTextAlignment( ALIGN_CENTER, ALIGN_BOTTOM );
plot.DrawText( cityLon, cityLat, cityNames );

/* draw reference axis showing scale in miles */
plot.DrawAxis(33.5, 54.25, 37, 54.25, {0 0.33 0.66 1},
     {'0' '25' '50' '75'}, {0.2 0.4 0.6 0.8}, 90 );
plot.DrawSetTextSize( 8 );
plot.DrawText( 35.25, 54.45, "Lieues communes de France" );

/* draw simple legend */
plot.DrawSetTextAlignment( ALIGN_LEFT, ALIGN_CENTER );
plot.DrawSetPenColor( BLACK );
plot.DrawSetBrushColor( RED );
plot.DrawRectangle( 34.6, 54.76, 35, 54.84, true );
plot.DrawText( 35.2, 54.8, "Advance" );
plot.DrawSetBrushColor( GREY );
plot.DrawRectangle( 34.6, 54.66, 35, 54.74, true );
plot.DrawText( 35.2, 54.7, "Retreat" );

/* increase bottom margin and place temperature graphic within */
plot.GetAxisViewRange( YAXIS, ymin, ymax );
plot.SetAxisTickRange( YAXIS, ymin, ymax );
boxMin = ymin - (ymax-ymin)/3;  /* decrease by 1/3 */
boxMax = ymin - (ymax-ymin)/20; /* and by 1/20     */

plot.SetAxisViewRange( YAXIS, boxMin, ymax );
plot.GetAxisViewRange( XAXIS, xmin, xmax );

tempMin = -30;
tempMax = 0;
tempInc = 10;
```

Simple? - SAS

```
/* draw bounding box and temperature axis */
plot.DrawRectangle( xmin, boxMin, xmax, boxMax );
plot.DrawNumericAxis( xmax, boxMin, xmax, boxMax, tempMin, tempMax, 4 );

/* scale temperature so we can plot in data coordinates */
temp = (retreatTemp - tempMin)/(tempMax-tempMin);
temp = temp * ( boxMax - boxMin ) + boxMin;

/* draw temperature versus longitude */
plot.DrawSetPenWidth( 2 );
plot.DrawSetPenStyle( DASHED );
plot.DrawLine( retreatLon, temp );

/* label temperature axis */
plot.DrawSetTextAlignment( ALIGN_RIGHT, ALIGN_BOTTOM );
plot.DrawText( xmax, boxMax, "Temp (C)" );

/* label temperature plot with dates of retreat */
k = 1:4;
plot.DrawSetTextAlignment( ALIGN_CENTER, ALIGN_TOP );
plot.DrawText( retreatLon[k], temp[k], putn(retreatDate[k],'DATE5.') );
k = 5:nrow(retreatLon);
plot.DrawSetTextAlignment( ALIGN_CENTER, ALIGN_BOTTOM );
plot.DrawText( retreatLon[k], temp[k], putn(retreatDate[k],'DATE5.') );

plot.ActivateWindow();

start DefineData;
GREY = 0808080x;
/* city names and locations for annotations */
cityNames = {...};
cityPos = {...};
cityLon = cityPos[,1];
cityLat = cityPos[,2];
```

Simple? - SAS

```
/* temperature during retreat */
retreat = {...};
retreatLon  = retreat[,1];
retreatTemp = retreat[,2];
retreatDate = retreat[,3];

army = {...];
finish;

start PlotContinuousDataPath( mX, mY, mSize, Plot plot );
     x = rowvec( mX );
     y = rowvec( mY );
     size = rowvec( mSize );

     n = ncol(x);
     if n < 2 then do;
          print "GetContinuousPath: path must contain at least two vertices";
          abort;
          end;
     if n ^= ncol(y) | n ^= ncol(size) then do;
          print "GetContinuousPath: parameter arrays are different lengths";
          abort;
          end;

   /* translate all coordinates into pixel coords */
   plot.GetGraphAreaWidthHeight( winWidth, winHeight );

   /* find pixel coordinates of frame */
   plot.GetGraphAreaMargins( leftGraphMargin, rightGraphMargin,
                             topGraphMargin, bottomGraphMargin );

   plotHeight  = (1 - topGraphMargin - bottomGraphMargin) * winHeight;
   plotWidth   = (1 - leftGraphMargin - rightGraphMargin) * winWidth;
   plotLeft    = leftGraphMargin * winWidth;
   plotRight   = plotLeft + plotWidth;
   plotBottom  = bottomGraphMargin * winHeight;
   plotTop     = plotBottom + plotHeight;
```

SYSTAT

Simple? - SAS

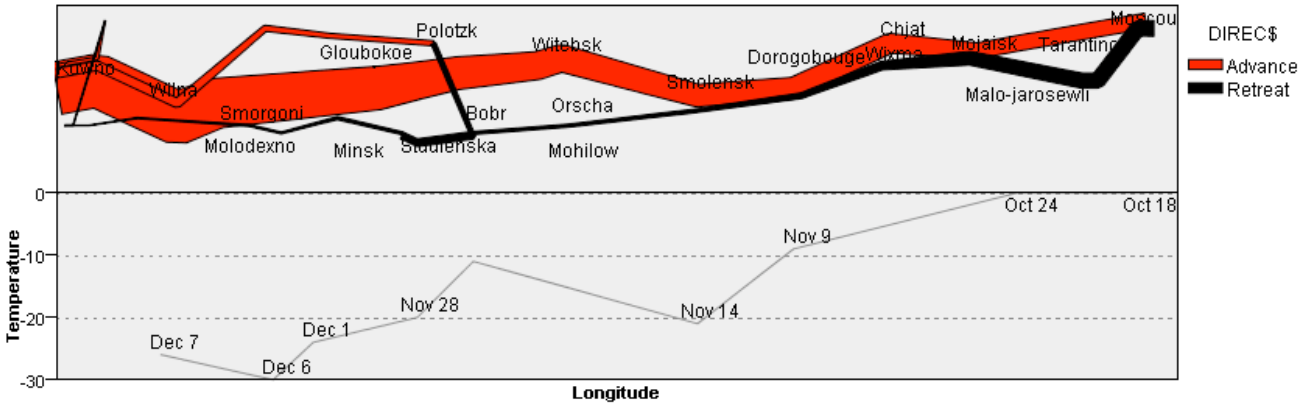
```
/* find pixel coordinates of plot area (includes plot margins) */
plot.GetPlotAreaMargins( leftPlotMargin, rightPlotMargin,
                         topPlotMargin, bottomPlotMargin );
drawHeight  = (1 - topPlotMargin - bottomPlotMargin) * plotHeight;
drawWidth   = (1 - leftPlotMargin - rightPlotMargin) * plotWidth;
drawLeft    = plotLeft + leftPlotMargin * plotWidth;
drawRight   = drawLeft + drawWidth;
drawBottom  = plotBottom + bottomPlotMargin * plotHeight;
drawTop     = drawBottom + drawHeight;

run GetPlotAreaDataCoordinates( plotXMin, plotXMax, plotYMin, plotYMax, plot );

  /* translate data into [plot area] pixels */
  x = plotLeft   + (plotRight-plotLeft)/(plotXMax-plotXMin) * (x - plotXMin);
  y = plotBottom + (plotTop-plotBottom)/(plotYMax-plotYMin) * (y - plotYMin);

path = j(2*n, 2, .);

/* find points that define width of first line segment */
u = y[1]-y[2] || x[2]-x[1];
uNorm = u / sqrt(ssq(u)); /* unit length */
  wx = size[1]/2 * uNorm[1];
  wy = size[1]/2 * uNorm[2];
  path[1,1] = x[1] + wx;
  path[1,2] = y[1] + wy;
  path[2*n,1] = x[1] - wx;
  path[2*n,2] = y[1] - wy;
```

SYSTAT

Simple? - SAS

```
   /* find points that define width of last line segment */
   u = y[n-1]-y[n] || x[n]-x[n-1];
   uNorm = u / sqrt(ssq(u)); /* unit length */
   wx = size[n]/2 * uNorm[1];
   wy = size[n]/2 * uNorm[2];
   path[n,1] = x[n] + wx;
   path[n,2] = y[n] + wy;
   path[n+1,1] = x[n] - wx;
   path[n+1,2] = y[n] - wy;

   /* now convert back to data coordinates */
   path[,1] = plotXMin + (plotXMax-plotXMin)/(plotRight-plotLeft) * (path[,1] - plotLeft);
   path[,2] = plotYMin + (plotYMax-plotYMin)/(plotTop-plotBottom) * (path[,2] - plotBottom);
   plot.DrawUseDataCoordinates();
   plot.DrawPolygon( path[,1], path[,2], true );
finish;
```

Simple - nViZn



Simple - nViZn

The blue statements are data definitions. The rest is the graphics program. This program executes in SPSS to produce the Minard graphic on the last slide.

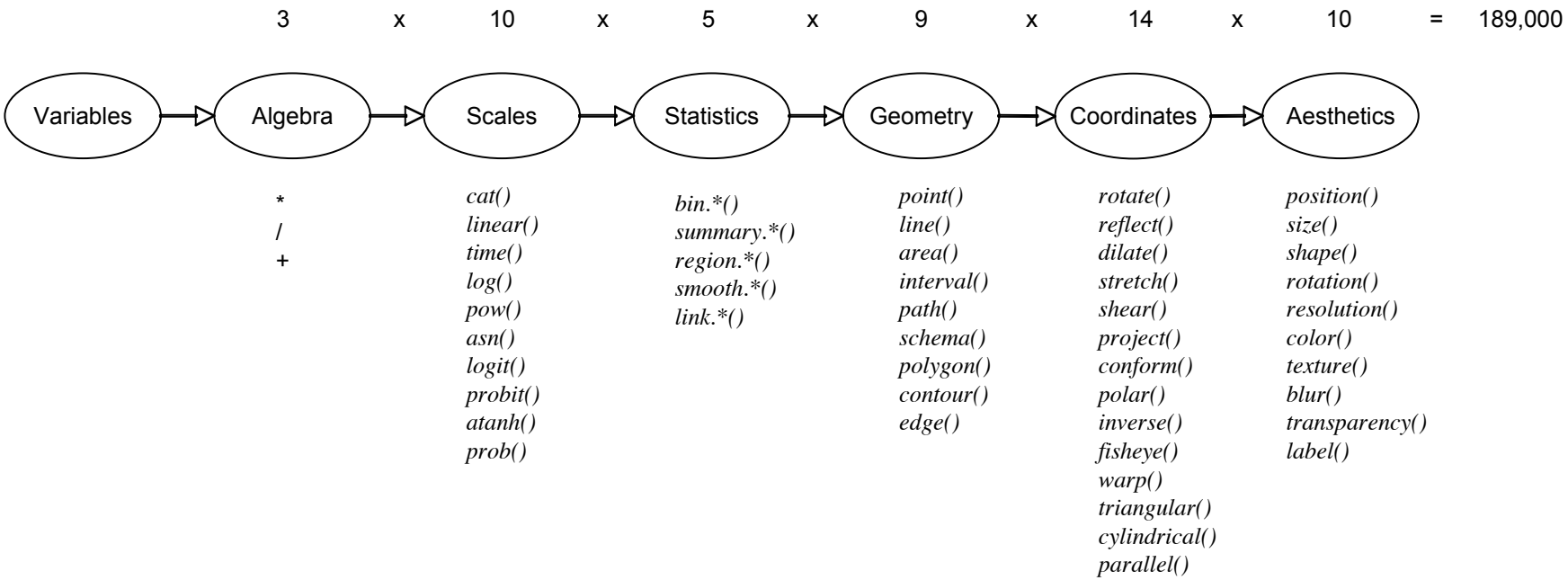
```
SOURCE: troops = csvSource(file("../data/minard troops.csv"))
SOURCE: cities = csvSource(file("../data/minard cities.csv"))
SOURCE: temperature = csvSource(file("../data/minard temperature.csv"))
DATA: lonp = col(source(troops), name("LON"))
DATA: latp = col(source(troops), name("LAT"))
DATA: surviv = col(source(troops), name("SURVIV"))
DATA: direc = col(source(troops), name("DIREC$"), unit.category())
DATA: division = col(source(troops), name("DIVISION$"), unit.category())
DATA: latc = col(source(cities), name("LAT"))
DATA: lonc = col(source(cities), name("LON"))
DATA: city = col(source(cities), name("CITY$"), unit.category())
DATA: lont = col(source(temperature), name("LON"))
DATA: temp = col(source(temperature), name("TEMP"))
DATA: date = col(source(temperature), name("DATE$"), unit.category())

GRAPH: begin(origin(0.0px, 0.0px), scale(720.0px, 120.0px))
GUIDE: axis(dim(1), null())
GUIDE: axis(dim(2), null())
GUIDE: legend(aesthetic(aesthetic.size), null())
SCALE: cat(aesthetic(aesthetic.color), map(("Advance",color.red), ("Retreat",color.black)))
SCALE: linear(aesthetic(aesthetic.size), aestheticMinimum(size."0px"), aestheticMaximum(size."20px"))
ELEMENT: path(position(lonp*latp), size(surviv), color(direc), split(division))
ELEMENT: point(position(lonc*latc), size(size."0px"), label(city))
GRAPH: end()

GRAPH: begin(origin(0.0px, 120.0px), scale(720.0px, 120.0px))
GUIDE: axis(dim(1), label("Longitude"), ticks(null()))
GUIDE: axis(dim(2), label("Temperature"), start(0.0), delta(10.0), gridlines())
ELEMENT: path(position(lont*temp), color(color.gray))
ELEMENT: point(position(lont*temp), label(date), size(size."0px"))
GRAPH: end()
```

Expressive

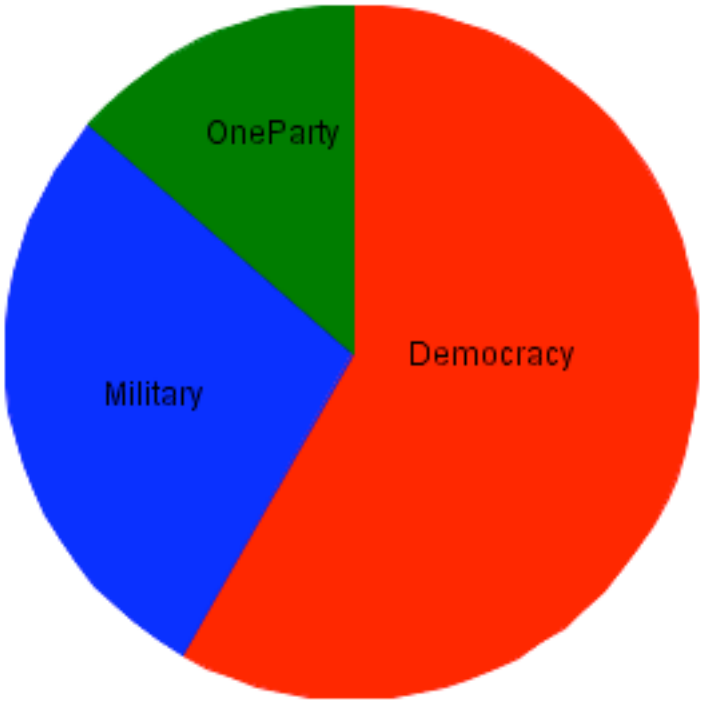
- Expressive means having a large results set.
- The following examples suggest the range of results the GoG model can produce.
- The range of possibilities greatly exceeds the capabilities of graphics production libraries and toolkits such as VTK.

Expressive

One can get an idea of the magnitude of results by computing a product of methods:

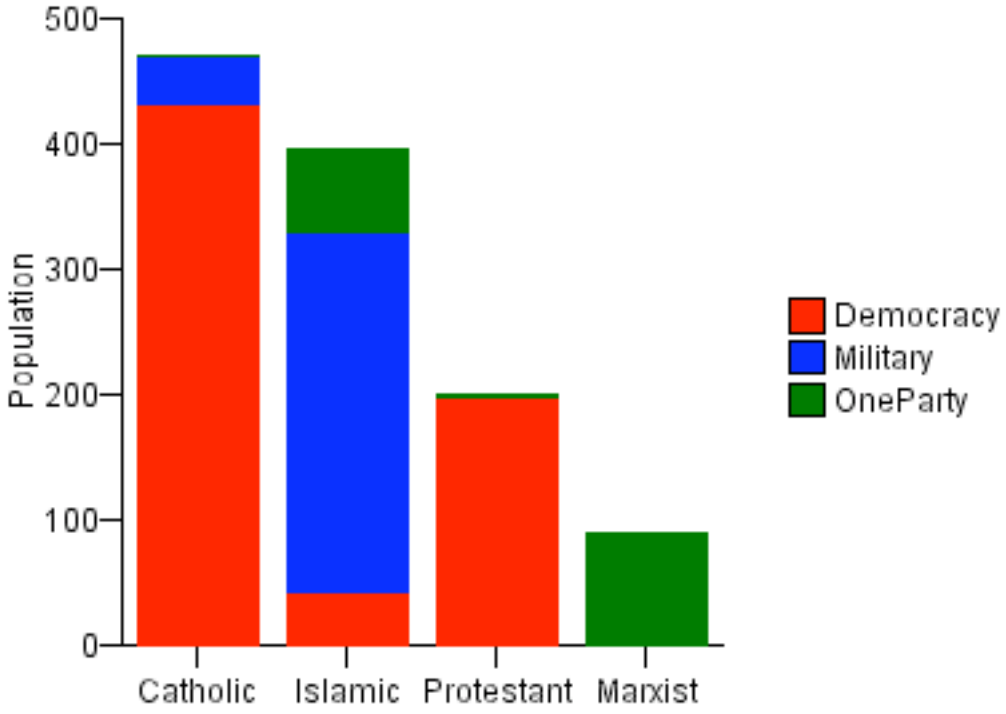


Expressive - some nViZn examples

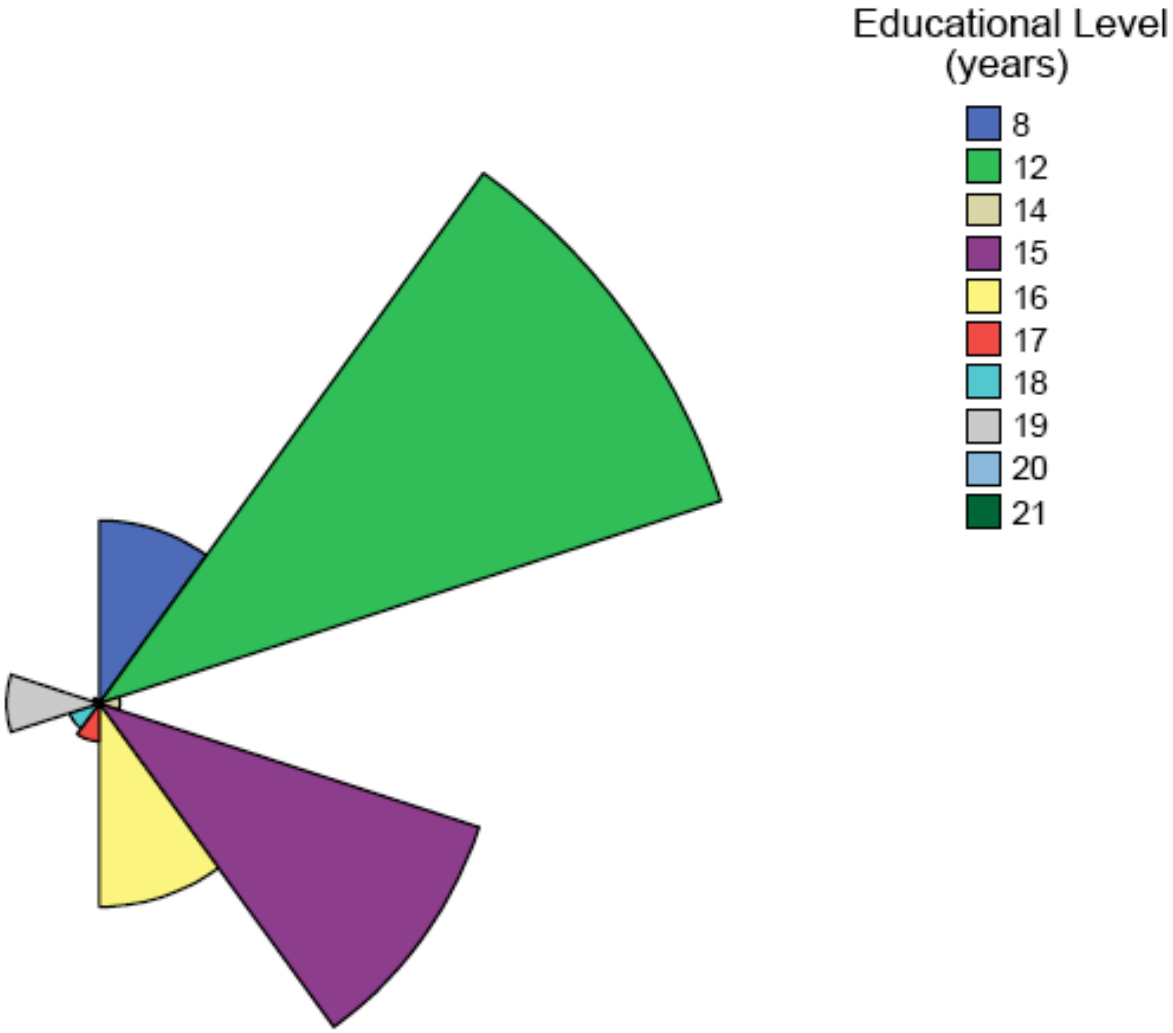


SYSTAT

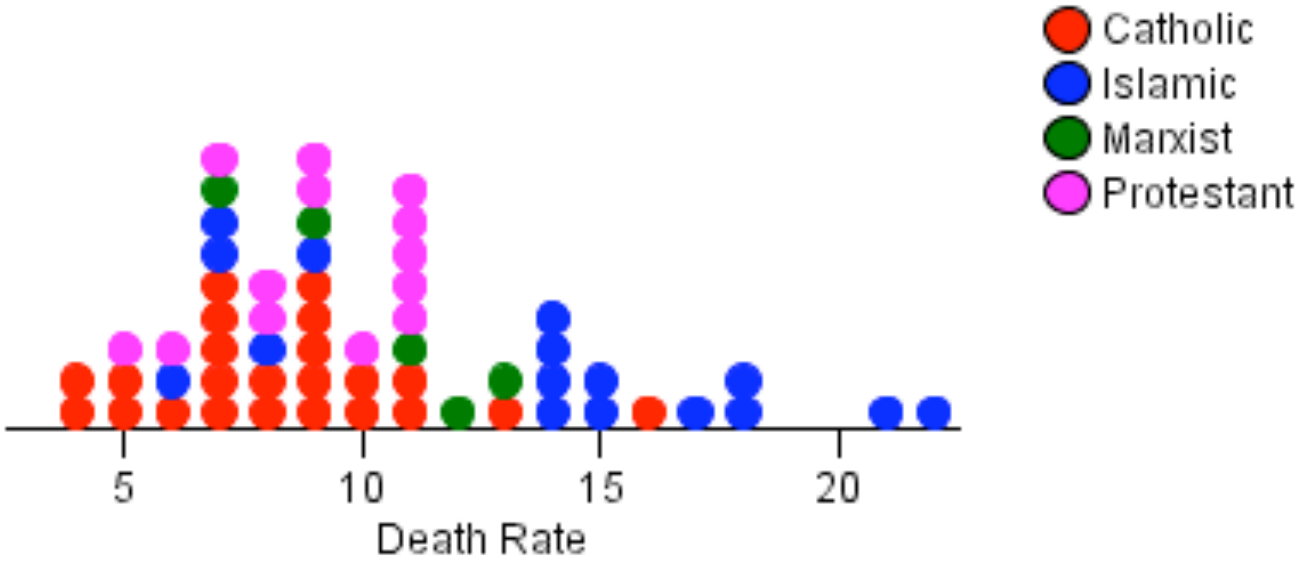
nViZn


500
400
300
200
100
0
Population
Catholic
Islamic
Protestant
Marxist
Democracy
Military
OneParty

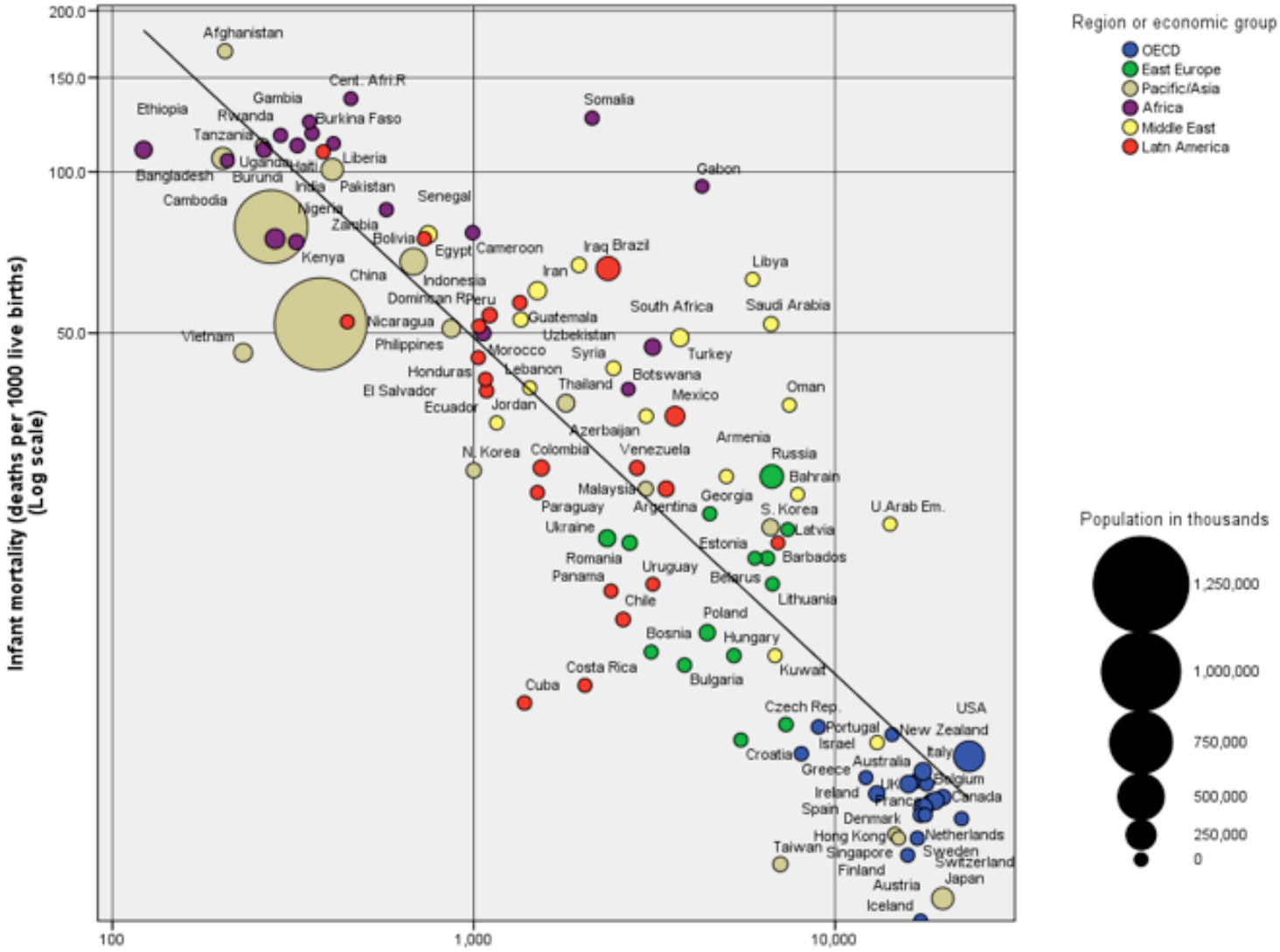
nViZn



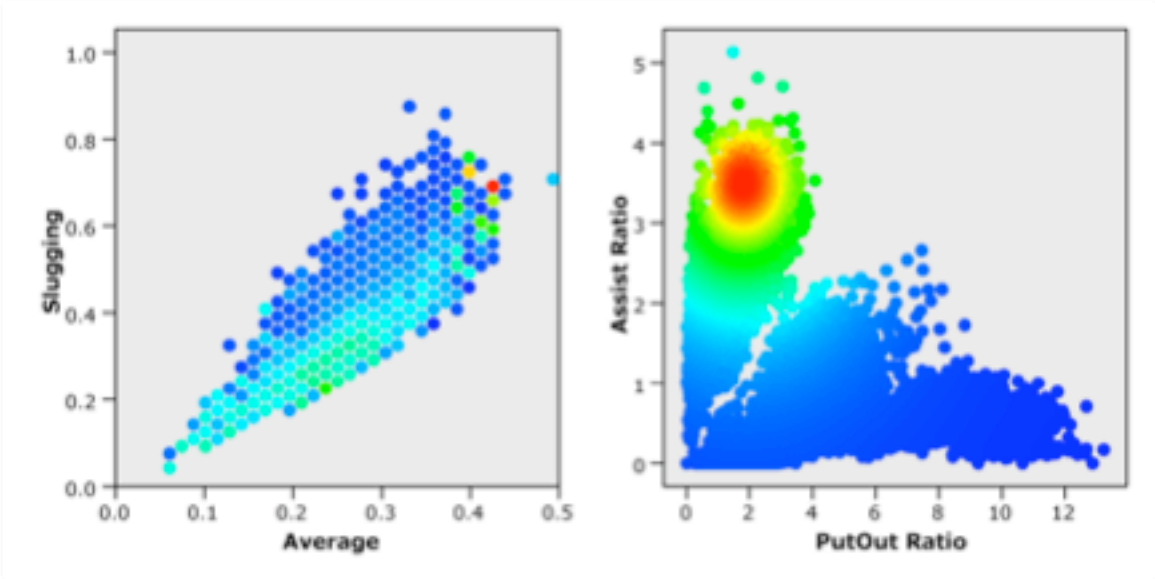
nViZn



nViZn

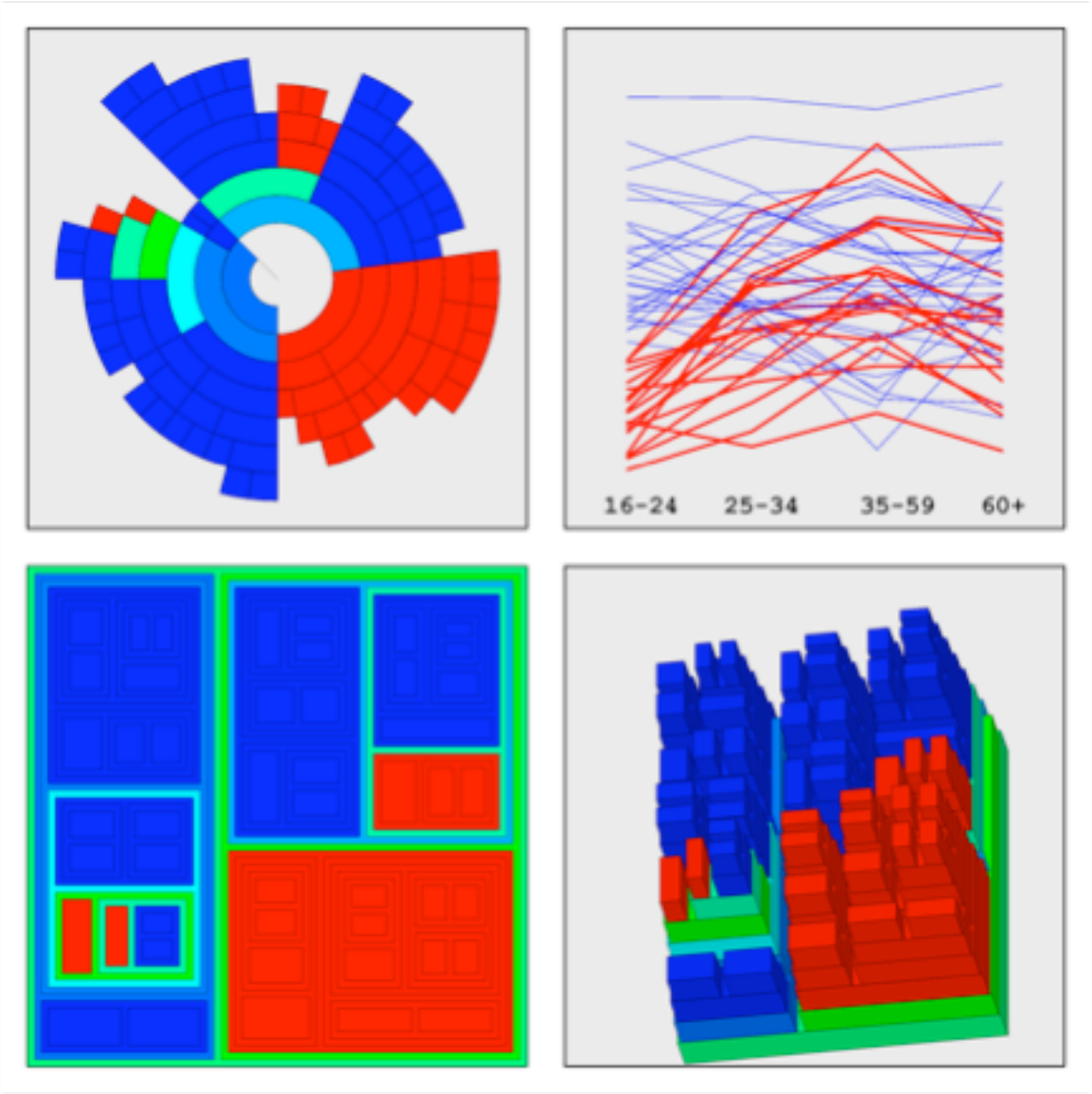

Infant mortality (deaths per 1000 live births) (Log scale)
200.0
150.0
100.0
50.0
100
1,000
10,000
Gross domestic product / capita (Log scale)
Afghanistan
Cent. Afri.R
Gambia
Ethiopia
Rwanda
Burkina Faso
Tanzania
Liberia
Bangladesh
Uganda
Haiti
Burundi
India
Pakistan
Cambodia
Nigeria
Senegal
Zambia
Bolivia
Egypt
Cameroon
Kenya
China
Indonesia
Dominican R
Peru
Nicaragua
Guatemala
Vietnam
Philippines
Uzbekistan
Morocco
Honduras
Lebanon
El Salvador
Thailand
Ecuador
Jordan
Azerbaijan
N. Korea
Colombia
Venezuela
Malaysia
Paraguay
Argentina
Ukraine
Romania
Panama
Uruguay
Chile
Poland
Bosnia
Hungary
Costa Rica
Bulgaria
Cuba
Somalia
Gabon
Iraq
Brazil
Iran
Libya
South Africa
Saudi Arabia
Turkey
Syria
Botswana
Mexico
Oman
Armenia
Russia
Bahrain
Georgia
S. Korea
U.Arab Em.
Latvia
Estonia
Barbados
Belarus
Lithuania
Kuwait
Czech Rep.
USA
Portugal
New Zealand
Israel
Italy
Croatia
Australia
Greece
Belgium
Ireland
Canada
Spain
Denmark
Hong Kong
Netherlands
Taiwan
Singapore
Sweden
Switzerland
Finland
Austria
Japan
Iceland
Region or economic group
OECD
East Europe
Pacific/Asia
Africa
Middle East
Latn America
Population in thousands
1,250,000
1,000,000
750,000
500,000
250,000
0

nViZn



SYSTAT

nViZn

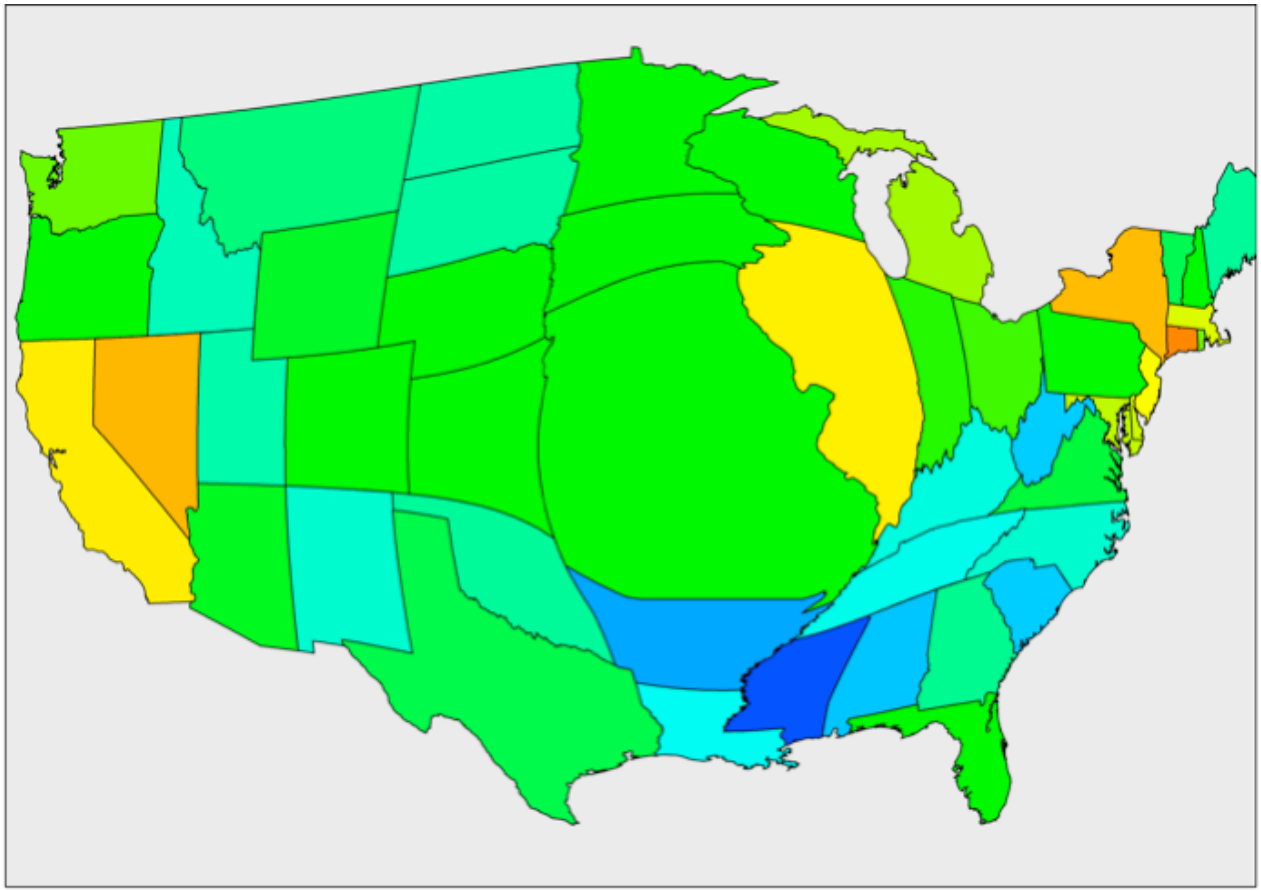


SYSTAT

nViZn

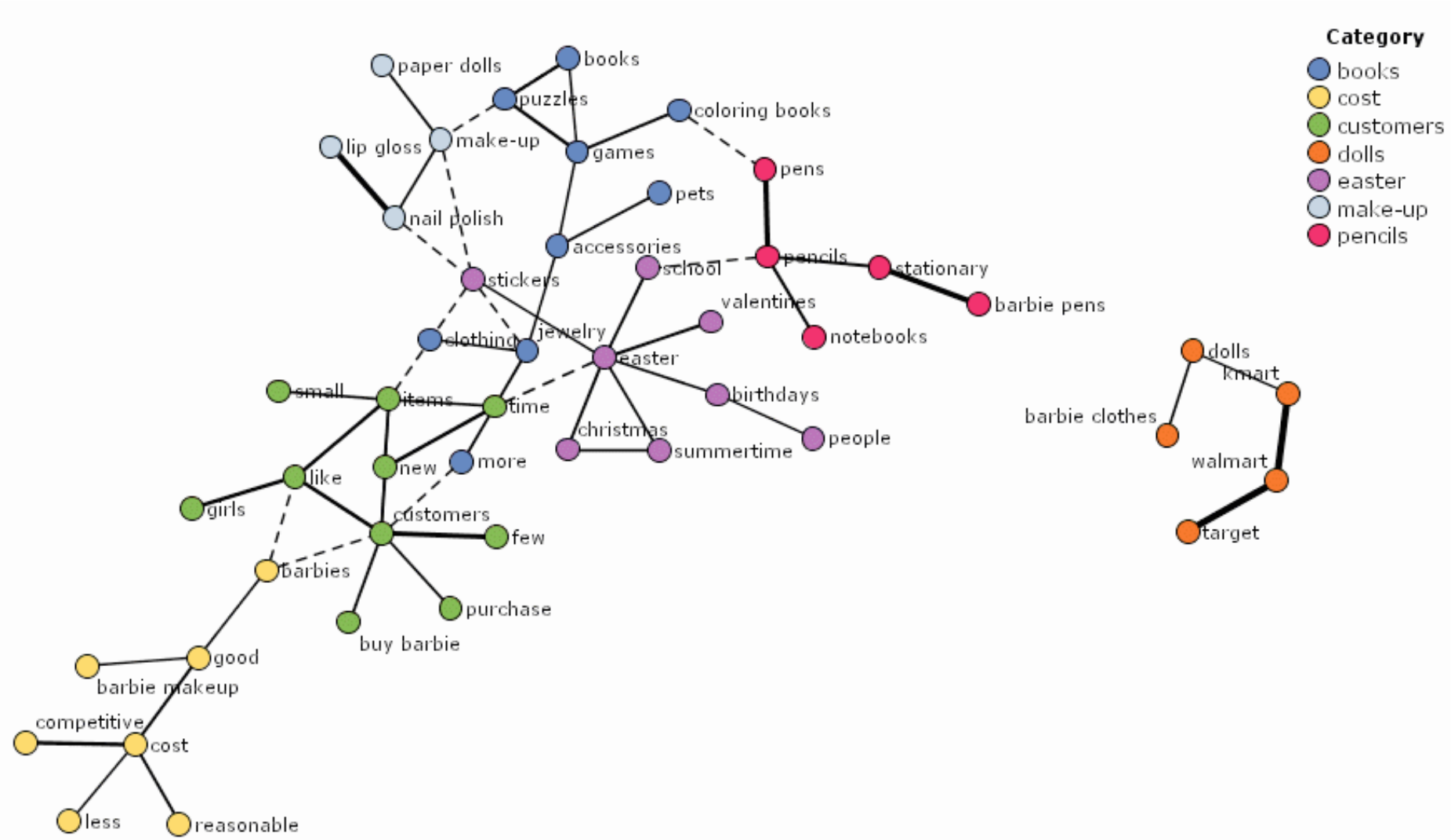

Holte 4%
Kilburn 6%
Amsterdam 4%
Woking 10%
Paris 2%
Rochester 23%
Arlington 1%
Chicago 37%
Xian 12%
Tokyo 1%

nViZn



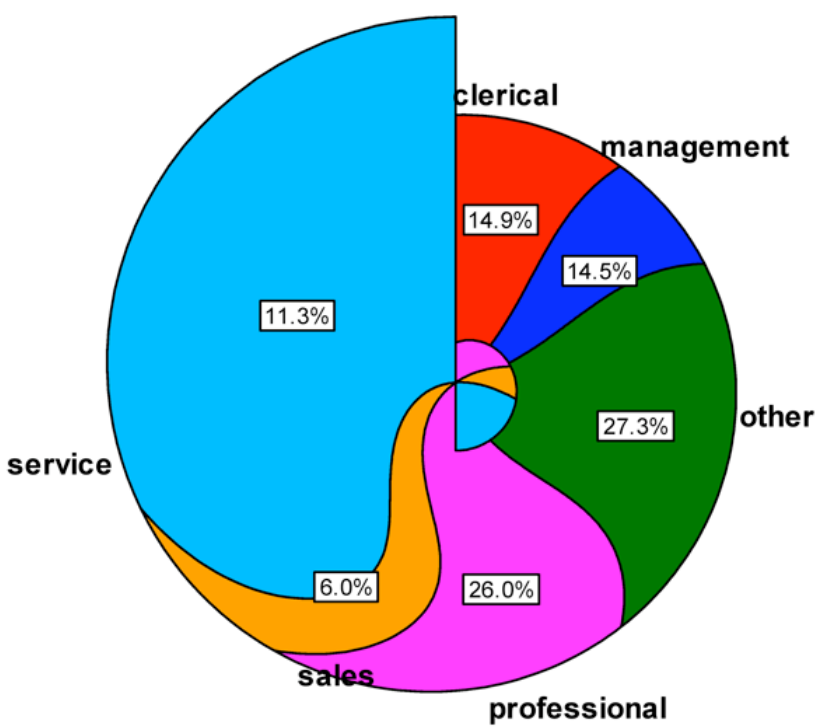
SYSTAT

nViZn


Category
books
cost
customers
dolls
easter
make-up
pencils
paper dolls
books
puzzles
coloring books
lip gloss
make-up
games
pens
pets
nail polish
accessories
pencils
stationary
school
stickers
barbie pens
valentines
clothing
jewelry
notebooks
easter
small
items
time
birthdays
christmas
people
summertime
more
new
like
girls
customers
few
barbies
purchase
buy barbie
good
barbie makeup
competitive
cost
less
reasonable
dolls
kmart
barbie clothes
walmart
target

nViZn

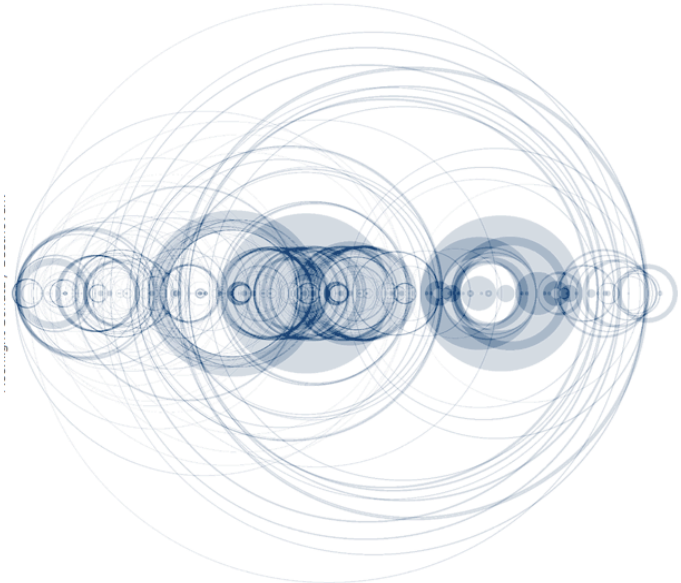
Expressive does not mean beautiful or useful. Here is a fisheye transformation that was accidentally applied to a pie chart by a programmer. It is meaningful but ugly.

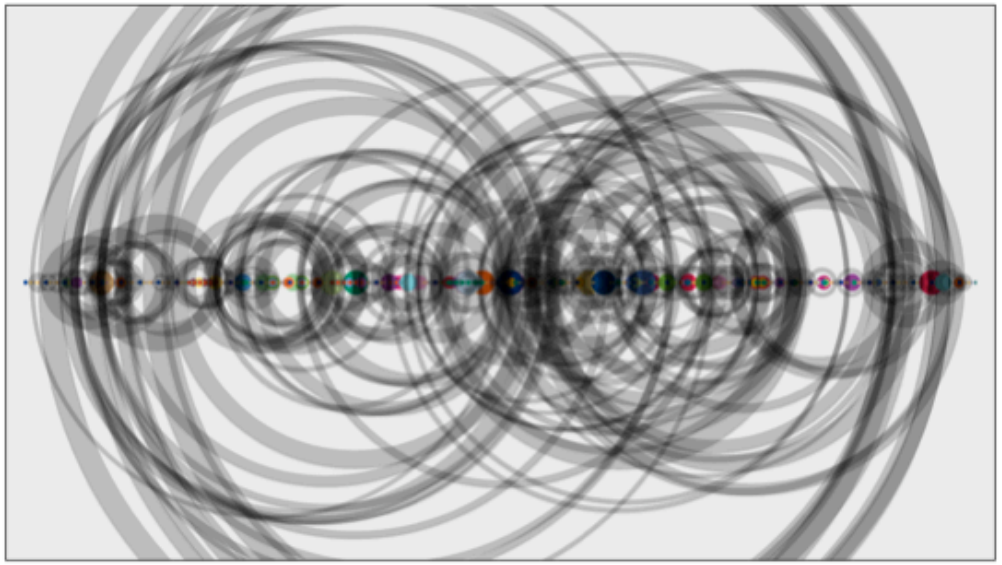


SYSTAT

nViZn

a cover tribute to Martin Wattenberg by Graham Wills
left: Wattenberg visualizing music; right: Wills visualizing his own lecture.
Wills programmed his example in nViZn the evening before his talk at the 2008 Joint Statistical Meetings (the day after Wattenberg's talk).

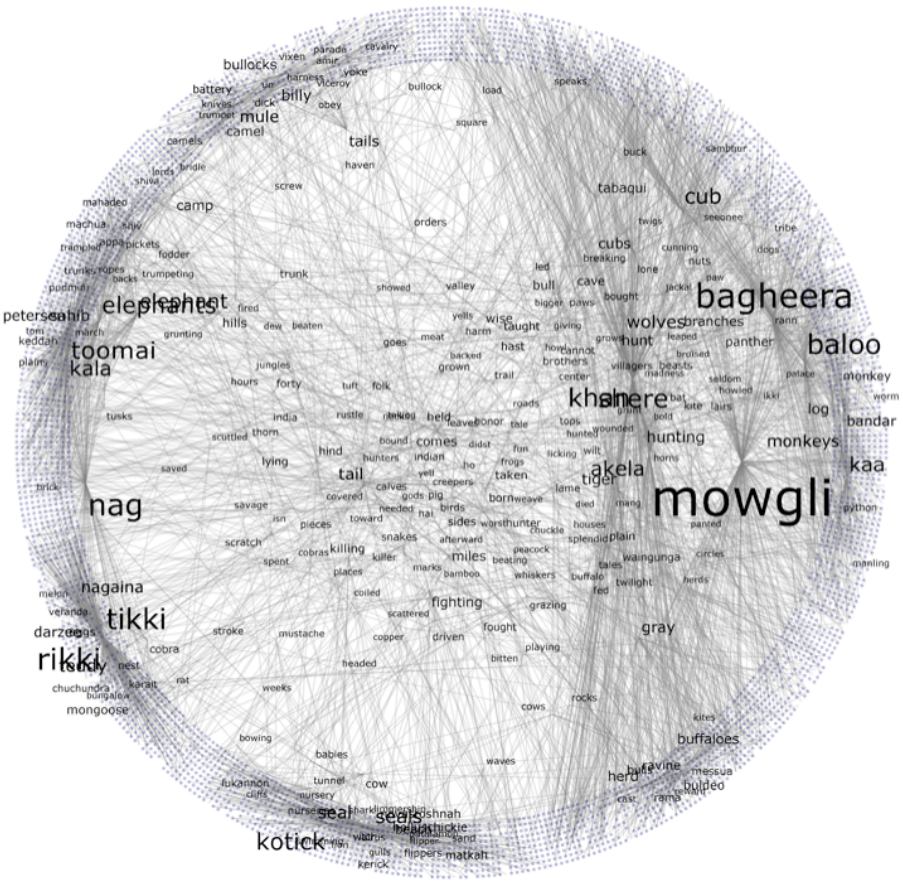




SYSTAT

nViZn

a cover tribute to Brad Paley by Graham Wills (drawn by nViZn)
left: Paley visualizing Alice in Wonderland; right: Wills visualizing Jungle Book.
Wills programmed his example in nViZn the evening before his talk at the 2008 Joint Statistical Meetings.

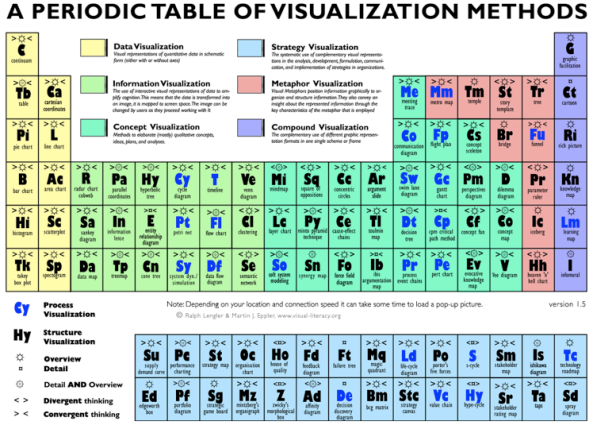


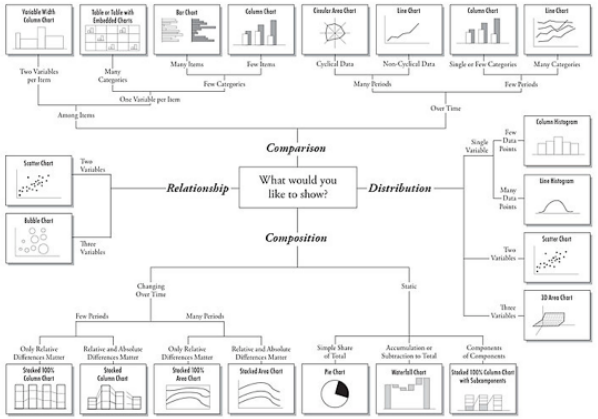
Coherent

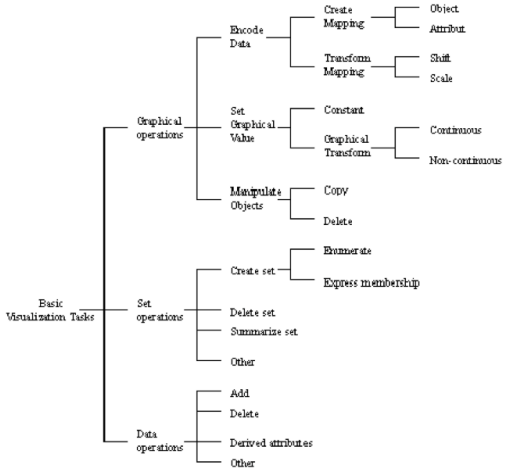
- Coherence comes from a grammar that does not depend on surface appearance.
- Taxonomies of visualization are frequently based on surface appearance.
- Taxonomies can be harmful because they mislead programmers, mislead designers, and lead everyone to think the world of graphics is complicated.

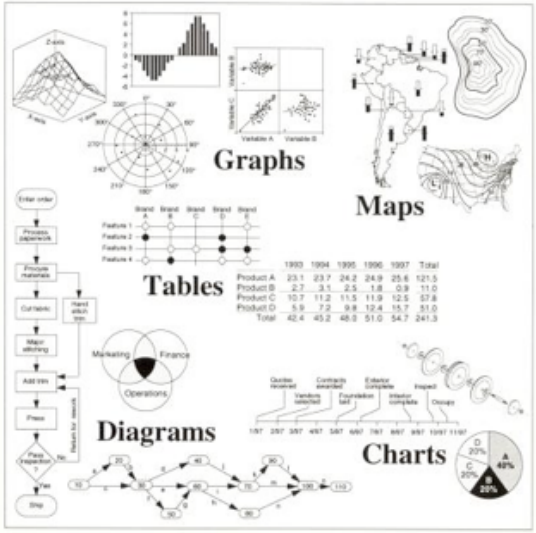
Coherent

- Taxonomies of charts are harmful.









Coherent

- Taxonomies of charts are harmful because...

this is similar to this

Coherent

- Taxonomies of charts are harmful because...

this is similar to this

Coherent

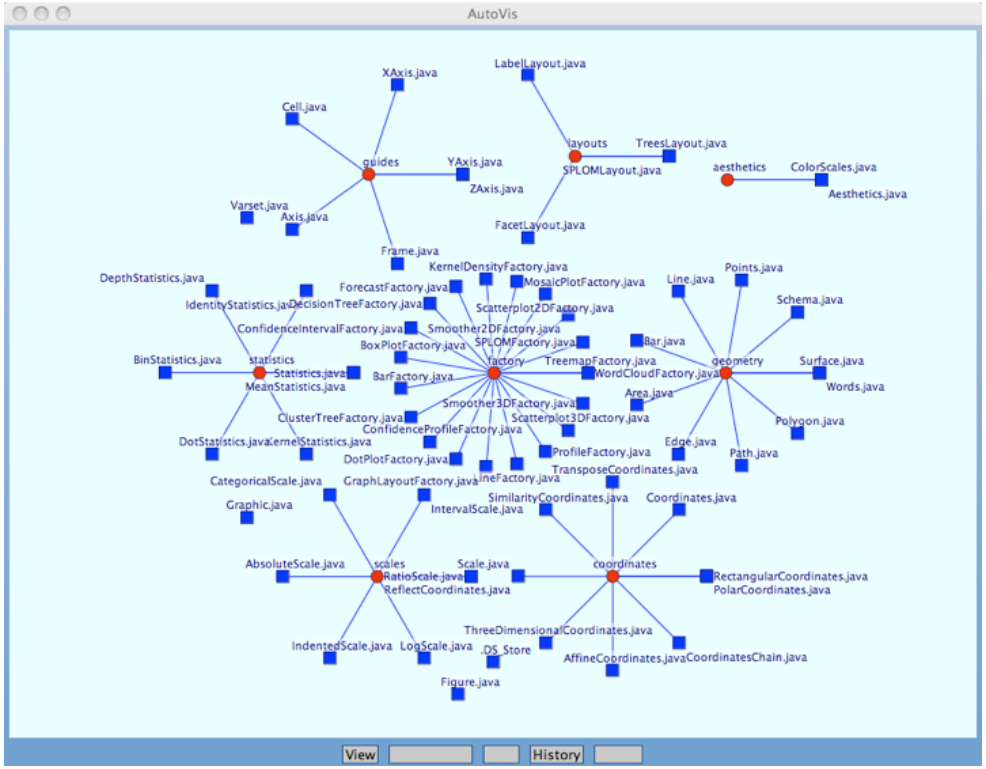
- Taxonomies of charts are harmful because...

but this

is *not* similar to this

Coherent

- Here is a graph layout of GoG Java packages.
- The Factory classes in the center contain automation methods for building graphics from the GoG surrounding classes.
- These factory methods all have a single coherent form.
- The next slide shows an example.



Coherent

- A treemap factory:

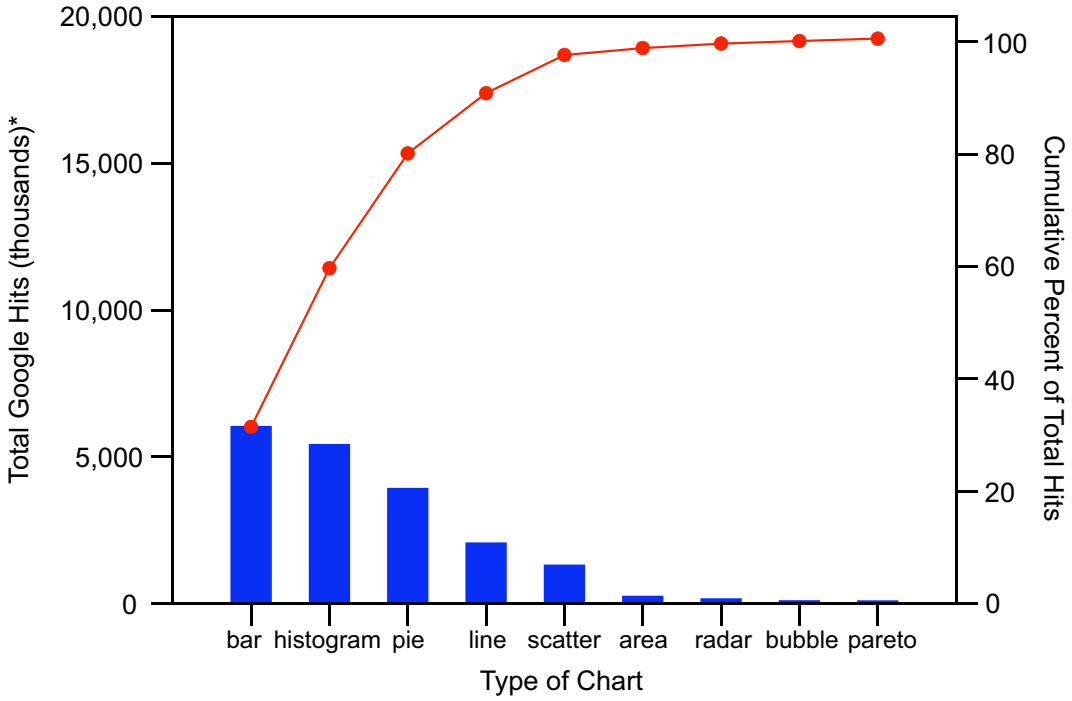
```
public abstract class TreemapFactory {
    public static Graphic build(FlatTable table, GraphicsStyles styles) {
        Renderer go = OutputFactory.makeBitmapGraphicsOutput(styles);
        Graphic graphic = new Graphic(go, styles, null);
        Scale[] scales = new Scale[]{new IntervalScale(table, 0)};
        Varset varset = new Varset(scales, table);
        Aesthetics aesthetics = new Aesthetics(varset);
        aesthetics.setPosition(new int[0]);
        aesthetics.setColor(new int[]{0});
        CoordinatesChain chain = new CoordinatesChain();
        Graphic branch = new Graphic(chain, graphic);
        new Polygon(aesthetics, branch, true);
        new Frame(aesthetics, branch);
        return graphic;
    }
}
```

Meaningful

- A meaningful language is based on identifying statements that are falsifiable.
- Graphics libraries and toolkits do not have this property.
- GoG asserts some popular charts are meaningless because they are ill-formed.
- To call these charts meaningful, defenders must falsify specific assumptions of GoG.
- To do so would lead to characterizing most charts as meaningless.

Meaningful

- GoG tells us the Pareto Chart is ill-formed



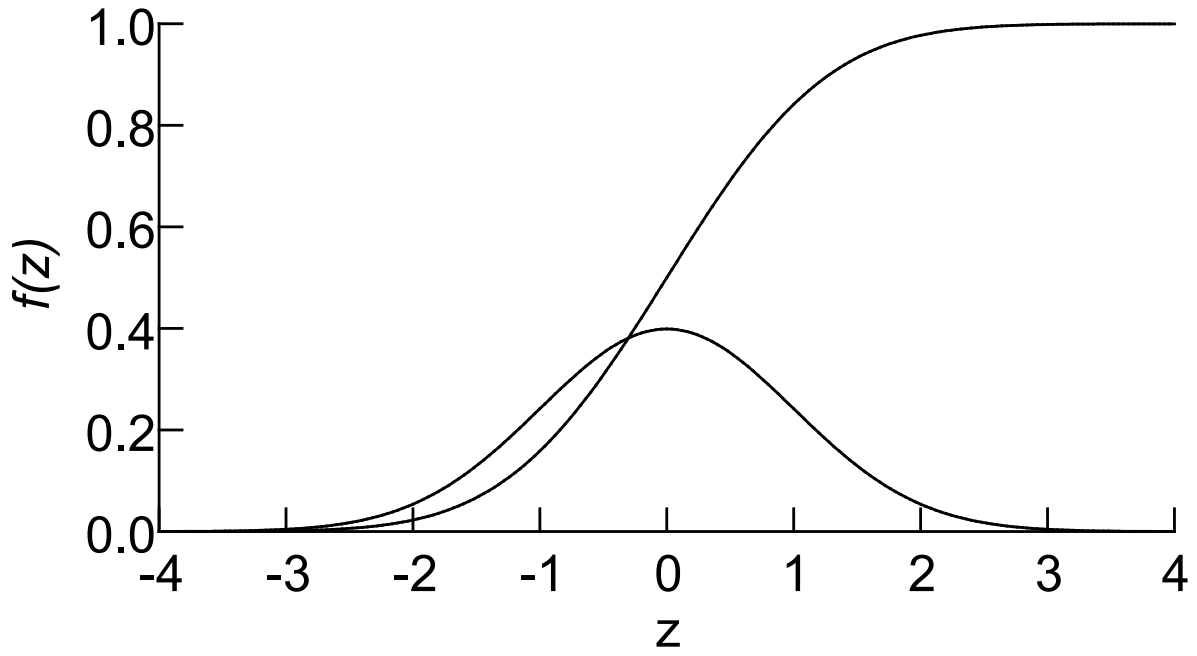
*queries = “<type>” + {“ chart”, ” graph”, ” plot”} plus {“scatterplot”, “histogram”}

This is a Pareto chart of charts.
It is based on total hits in Google searches (using an initial inventory derived from Google Sets).
On this basis, the Pareto chart is the ninth most popular chart in the world today.

SYSTAT

Meaningful

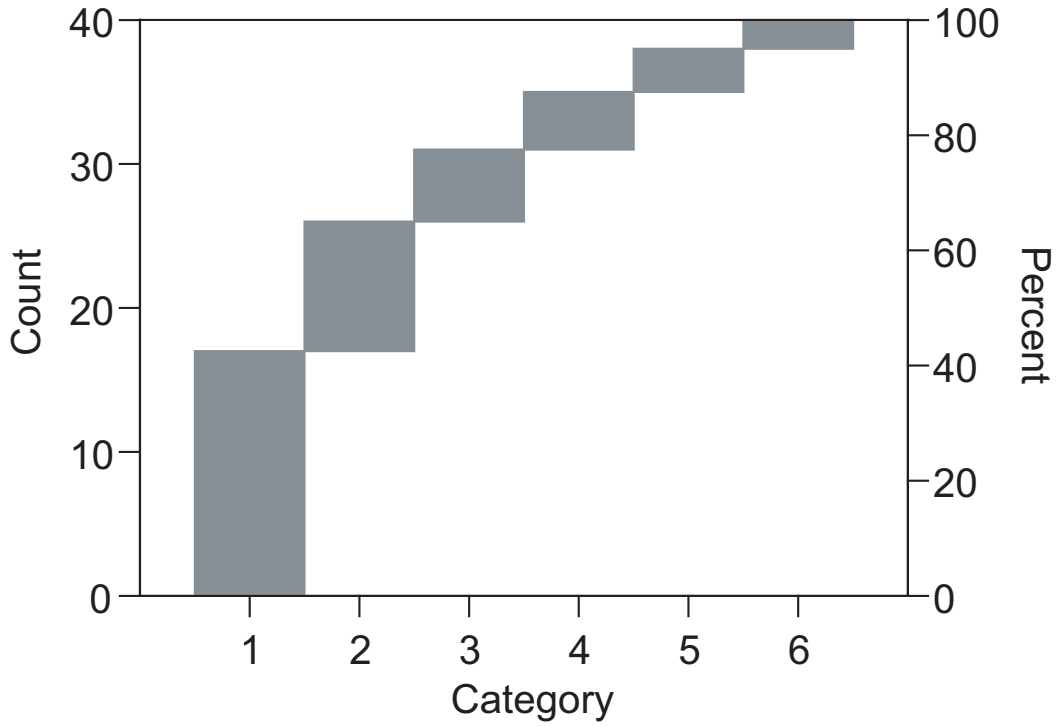
- GoG tells us this chart is ill-formed for the same reason.



The Pareto chart and this graph plot a density and cumulative distribution in the same frame.
It is nonsensical to plot a density and cumulative distribution in the same frame.
For the similar reasons, it is nonsensical to plot velocity and acceleration in the same frame.

Meaningful

Stacked Pareto Chart (Juran, 1951)



Juran invented the Pareto chart. This was his preferred version, however.
It is meaningful because the vertical scale is cumulative.
Unfortunately, the world chose the meaningless version.
Read more in Wilkinson, *The American Statistician*, 2006.

SYSTAT

Intelligent Graphics Systems

- AutoVis
- FASTAT

What is automated visualization?

- The computer looks at raw data and produces visualization(s) without coaching.
- The computer visually highlights aspects of data that are noteworthy.
- The computer presents data visualizations that help evaluate the appropriateness of a model.

What is the point of automated visualization?

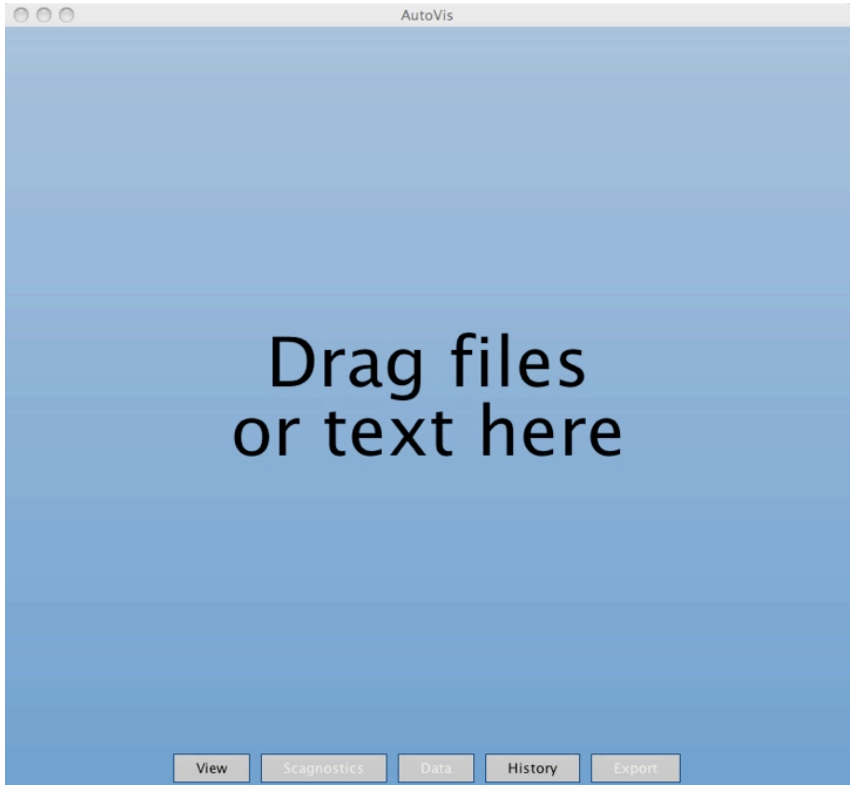
- Guided discovery
  - Data cleaning and catching anomalies
  - Checking assumptions
  - Controlling False Discovery Rate
- Data mining and machine learning
  - Classification
- NOT for deciding whether to buy Dubai real-estate or for submitting a paper to *Science*.

AutoVis

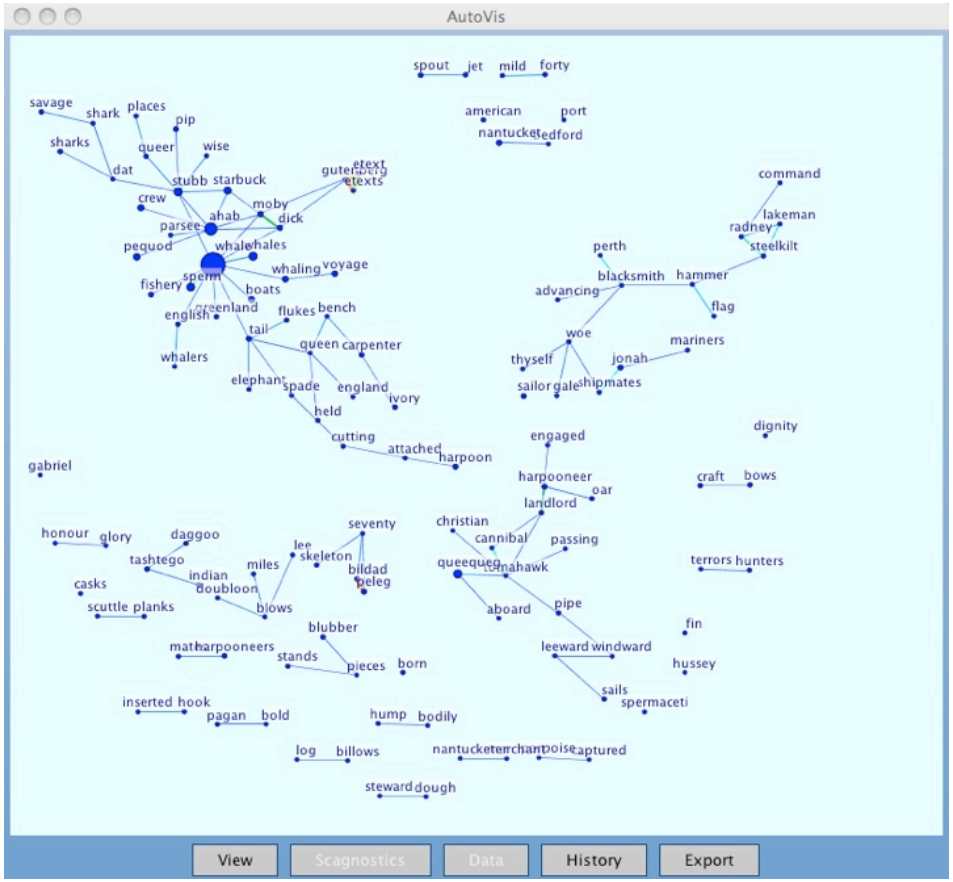
- A data viewer
  - Drag a file into window
  - Graphics appropriate to file structure
  - Uses scagnostics to find interesting views
  - Controls False Discovery Rate to prevent fishing expeditions

AutoVis

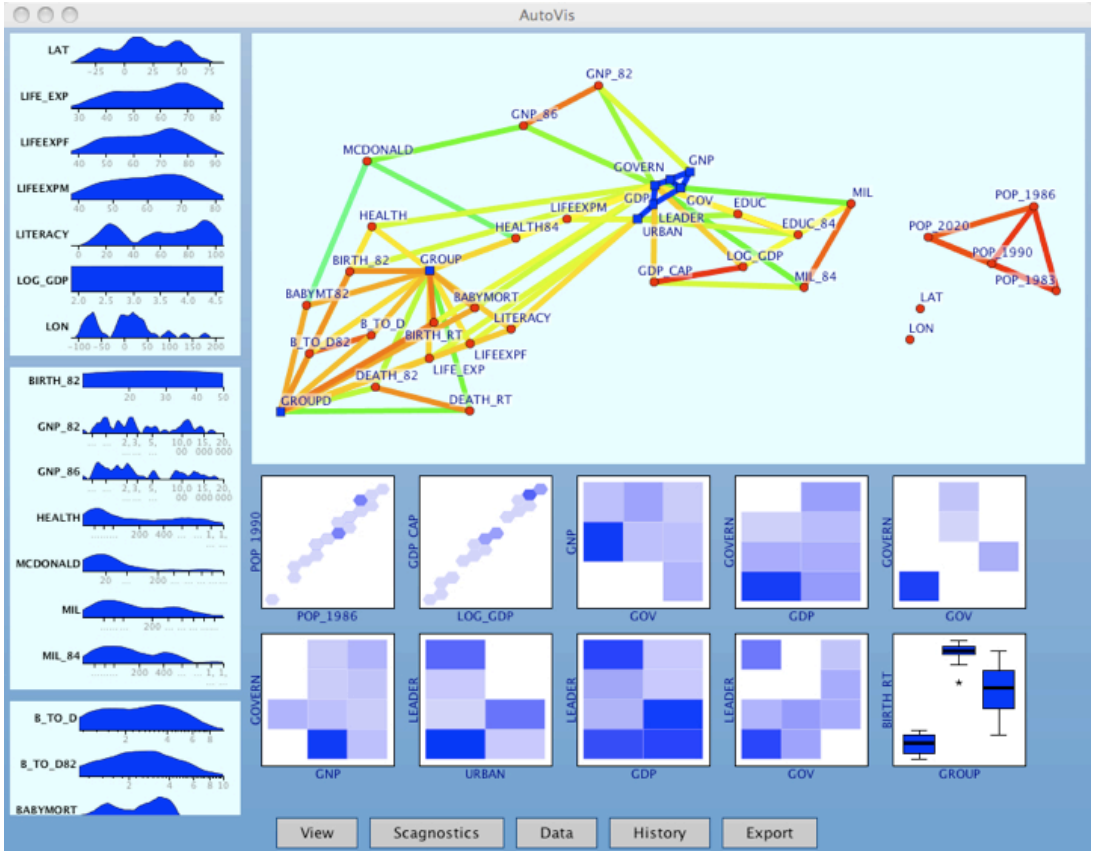
AutoVis has no menus or user manual. Grab a file from your local directory or from the Web and drag it in the window. The following slides show typical results.



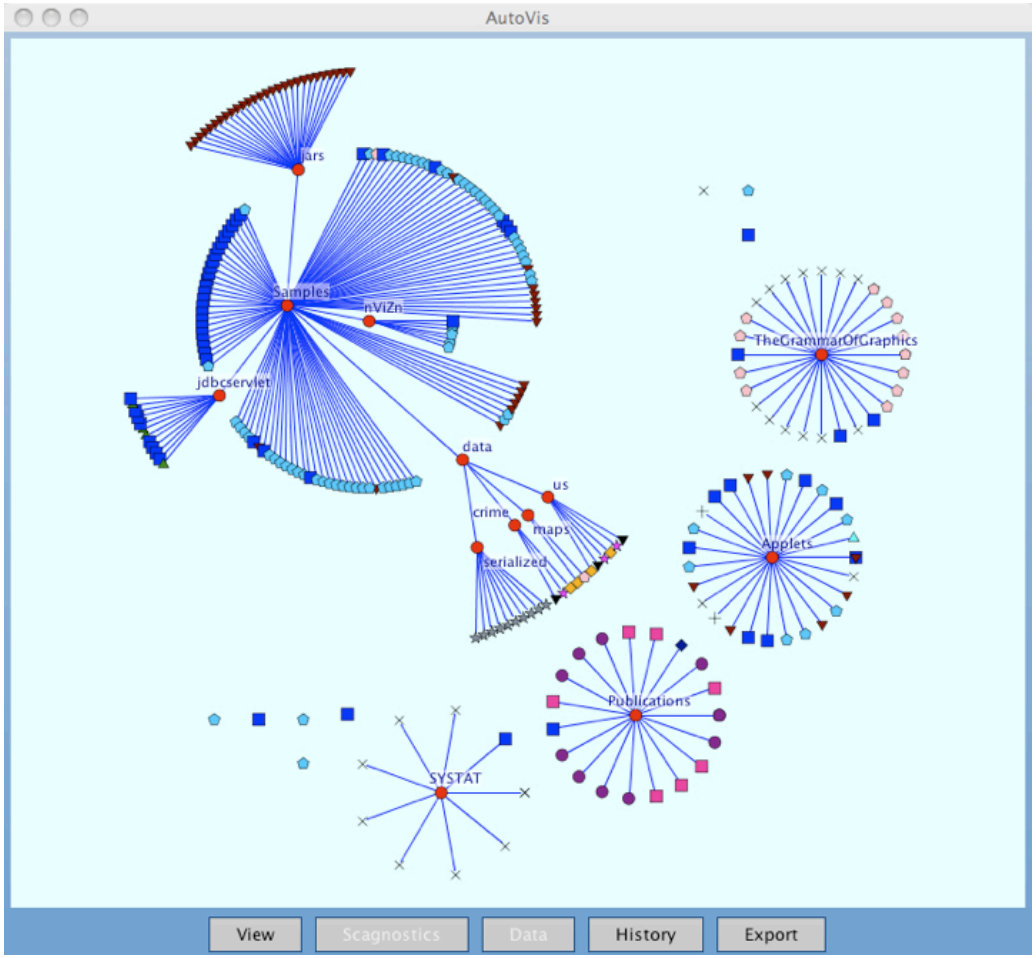
AutoVis


AutoVis
spout
jet
mild
forty
savage
shark
places
pip
sharks
queer
wise
dat
stubb
starbuck
crew
moby
ahab
dick
parsee
pequod
whale
whales
whaling
voyage
sperm
fishery
boats
english
greenland
flukes
bench
tail
queen
carpenter
whalers
elephant
spade
england
ivory
held
cutting
attached
harpoon
gabriel
american
port
nantucket
bedford
command
lakeman
radney
steelkilt
perth
blacksmith
hammer
advancing
flag
woe
mariners
thyself
jonah
sailor
gale
shipmates
dignity
engaged
harpooneer
oar
craft
bows
landlord
christian
cannibal
passing
queequeg
tomahawk
terrors
hunters
aboard
pipe
fin
leeward
windward
hussey
sails
spermaceti
honour
glory
daggoo
tashtego
indian
casks
doubloon
scuttle
planks
blows
seventy
lee
skeleton
miles
bildad
peleg
blubber
mat
harpooneers
stands
pieces
born
inserted
hook
pagan
bold
hump
bodily
log
billows
nantucketer
harpoise
captured
steward
dough
View
Scagnostics
Data
History
Export

AutoVis


AutoVis
LAT
LIFE_EXP
LIFEEXPF
LIFEEXPM
LITERACY
LOG_GDP
LON
BIRTH_82
GNP_82
GNP_86
HEALTH
MCDONALD
MIL
MIL_84
B_TO_D
B_TO_D82
BABYMORT
GNP_82
GNP_86
MCDONALD
GOVERN
GNP
HEALTH
HEALTH84
LIFEEXPM
GDP
GOV
EDUC
MIL
LEADER
URBAN
EDUC_84
LOG_GDP
GDP_CAP
MIL_84
BIRTH_82
GROUP
BABYMT82
BABYMORT
LITERACY
B_TO_D
B_TO_D82
BIRTH_RT
LIFEEXPF
LIFE_EXP
DEATH_82
GROUPD
DEATH_RT
POP_1986
POP_2020
POP_1990
POP_1983
LAT
LON
POP_1990
POP_1986
GDP_CAP
LOG_GDP
GNP
GOV
GOVERN
GDP
GOVERN
GOV
GOVERN
GNP
LEADER
URBAN
LEADER
GDP
LEADER
GOV
BIRTH_RT
GROUP
View
Scagnostics
Data
History
Export

AutoVis


AutoVis
jars
Samples
nViZn
jdbcservlet
data
us
crime
maps
serialized
TheGrammarOfGraphics
Applets
Publications
SYSTAT
View
Scagnostics
Data
History
Export

FASTAT

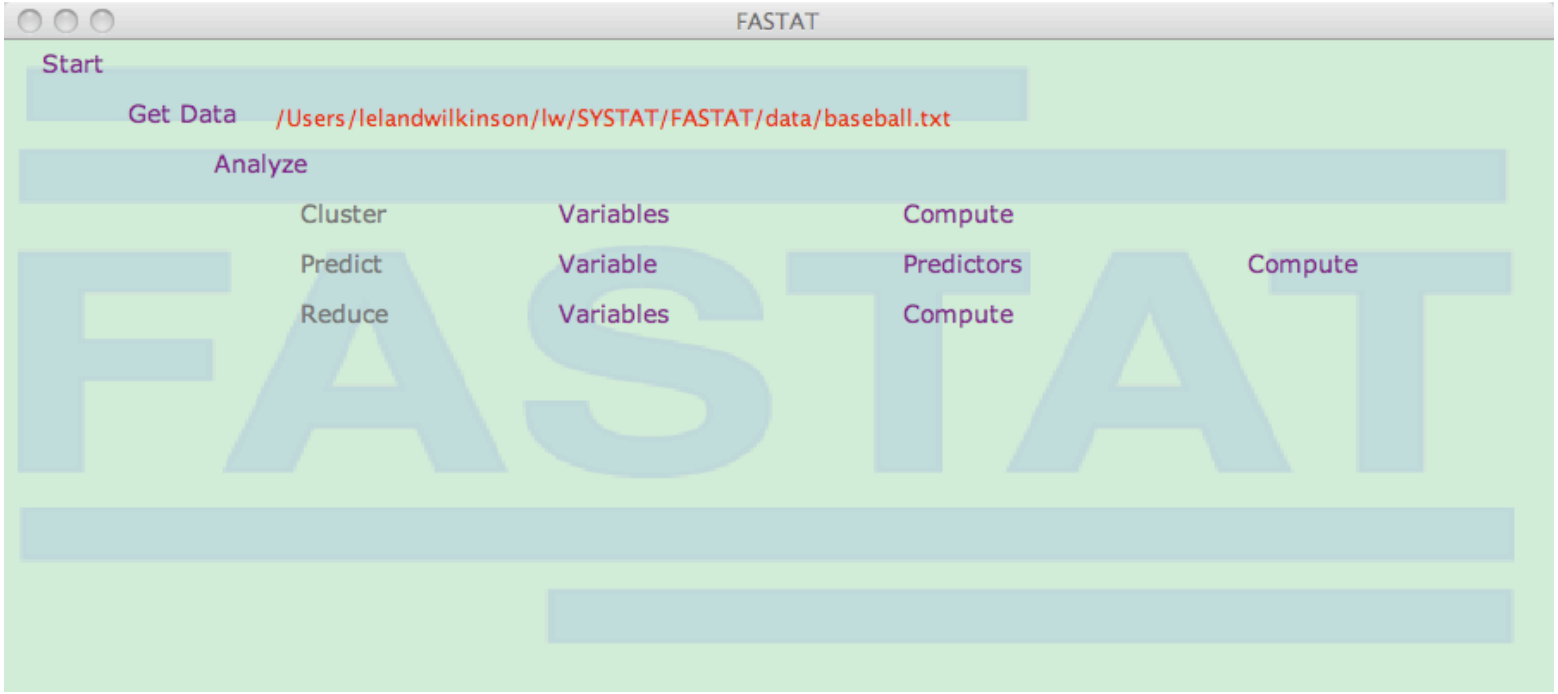
- A second opinion:
  - Look for outliers, distributions, coding errors...
  - Transform to meet assumptions
  - Converse with user
  - Produce annotated, hyperlinked output

FASTAT

FASTAT has no dialogs or user manual. There is no keyboard input. The main window fills up as the user chooses options by touching the purple words, as the following slide shows.

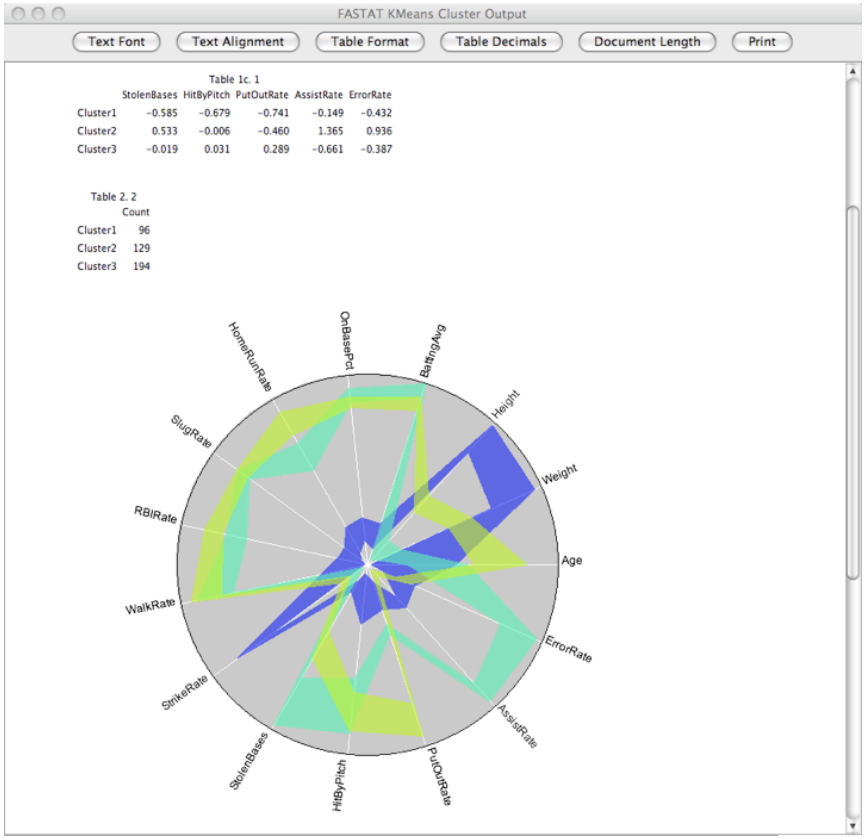
SYSTAT

FASTAT

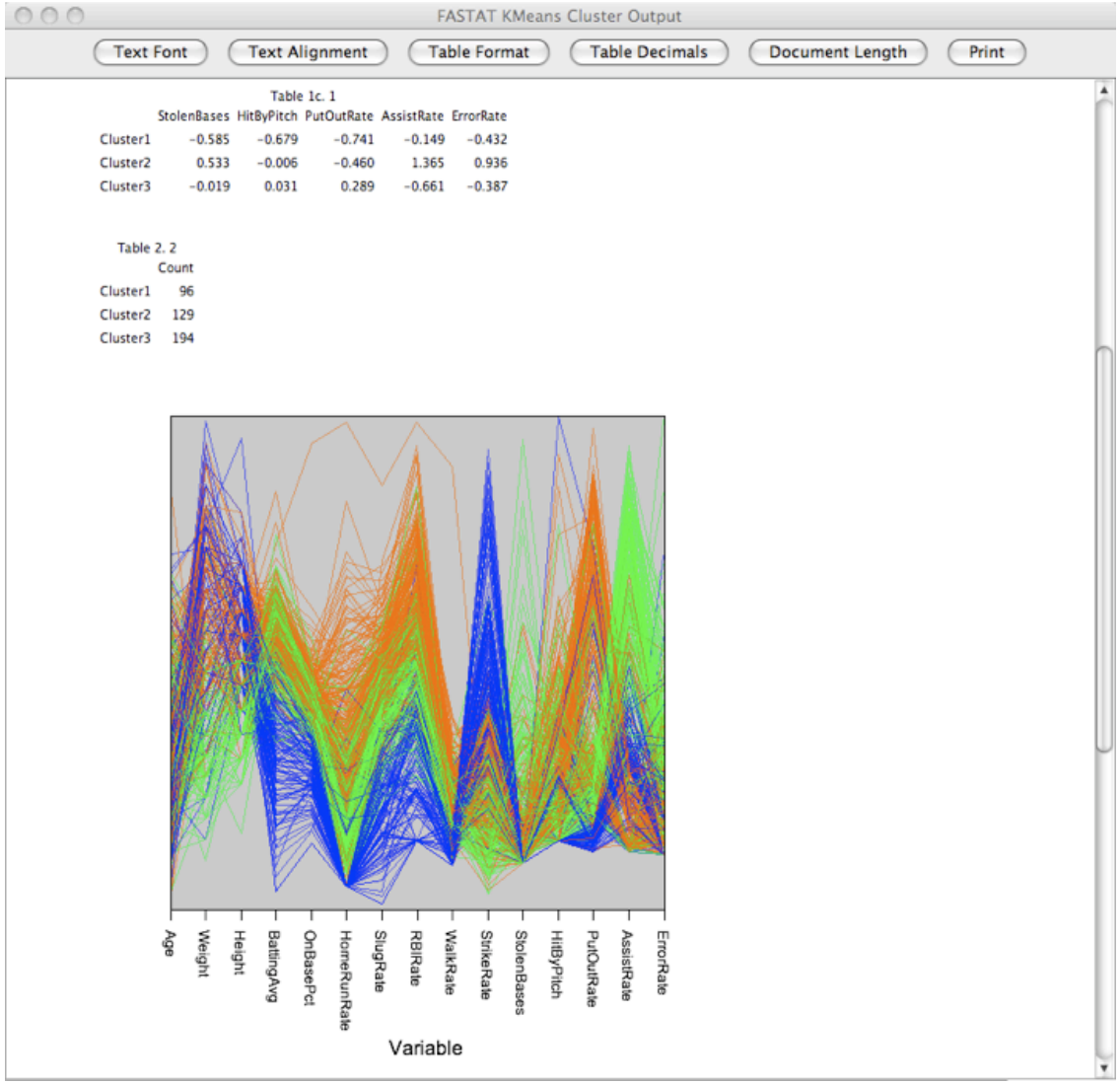

FASTAT
Start
Get Data
/Users/lelandwilkinson/lw/SYSTAT/FASTAT/data/baseball.txt
Analyze
Cluster
Variables
Compute
Predict
Variable
Predictors
Compute
Reduce
Variables
Compute
FASTAT

FASTAT

The output is a browser that contains graphics and text. Touching a graphic produces a popup that allows the user to change the form of the graphic in one tap, as the following slide shows.



FASTAT


FASTAT KMeans Cluster Output
Text Font
Text Alignment
Table Format
Table Decimals
Document Length
Print
Table 1c. 1
StolenBases HitByPitch PutOutRate AssistRate ErrorRate
Cluster1 -0.585 -0.679 -0.741 -0.149 -0.432
Cluster2 0.533 -0.006 -0.460 1.365 0.936
Cluster3 -0.019 0.031 0.289 -0.661 -0.387
Table 2. 2
Count
Cluster1 96
Cluster2 129
Cluster3 194
Age
Weight
Height
BattingAvg
OnBasePct
HomeRunRate
SlugRate
RBIRate
WalkRate
StrikeRate
StolenBases
HitByPitch
PutOutRate
AssistRate
ErrorRate
Variable

Conclusion

To build *intelligent* visual analytics, we need to understand the mathematical foundation of analytic visualizations.