ATOM: Unit Visualization Grammar

Deokgun Park*, Steven M. Drucker, and Roland Fernandez
*University of Maryland, College Park
Microsoft Research
intuinno@umd.edu

WHAT IS UNIT VISUALIZATION PROCESS?

WHAT IS UNIT VISUALIZATION PROCESS?
WHY IT IS IMPORTANT?

WHAT IS UNIT VISUALIZATION PROCESS?
WHY IT IS IMPORTANT?
HOW TO DRAW?

What is Unit Visualization?

Aggregated Visualization

	Group	Income
Bill Gates	A	60
Steve Jobs	A	30
John Doe	B	2
…	…	…

Aggregated Visualization

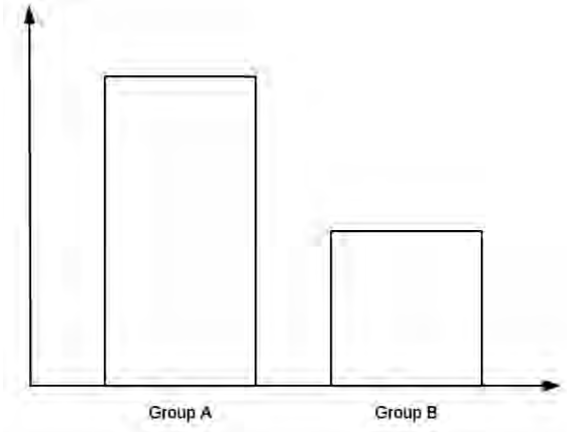
	Group	Income
Bill Gates	A	60
Steve Jobs	A	30
John Doe	B	2
…	…	…

	Total Income
Group A	60
Group B	30

Aggregated Visualization

	Group	Income
Bill Gates	A	60
Steve Jobs	A	30
John Doe	B	2
…	…	…

	Total Income
Group A	60
Group B	30

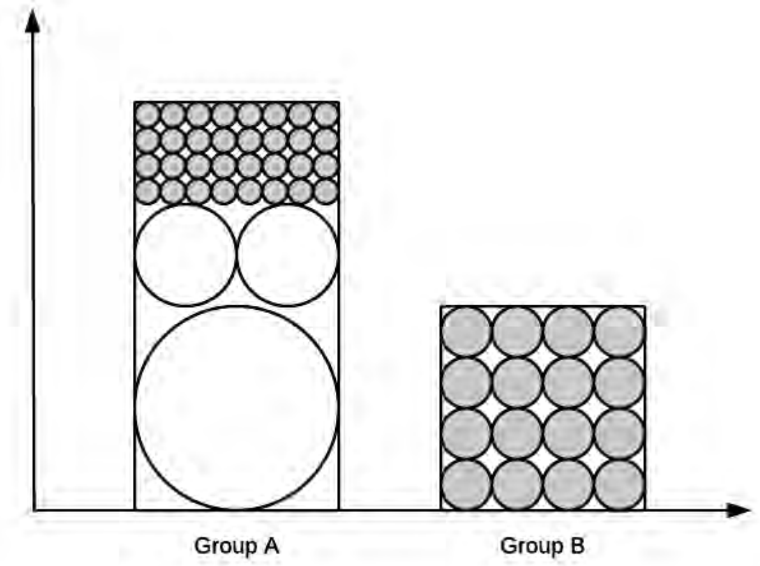


Unit Visualization

	Group	Income
Bill Gates	A	60
Steve Jobs	A	30
John Doe	B	2
…	…	…

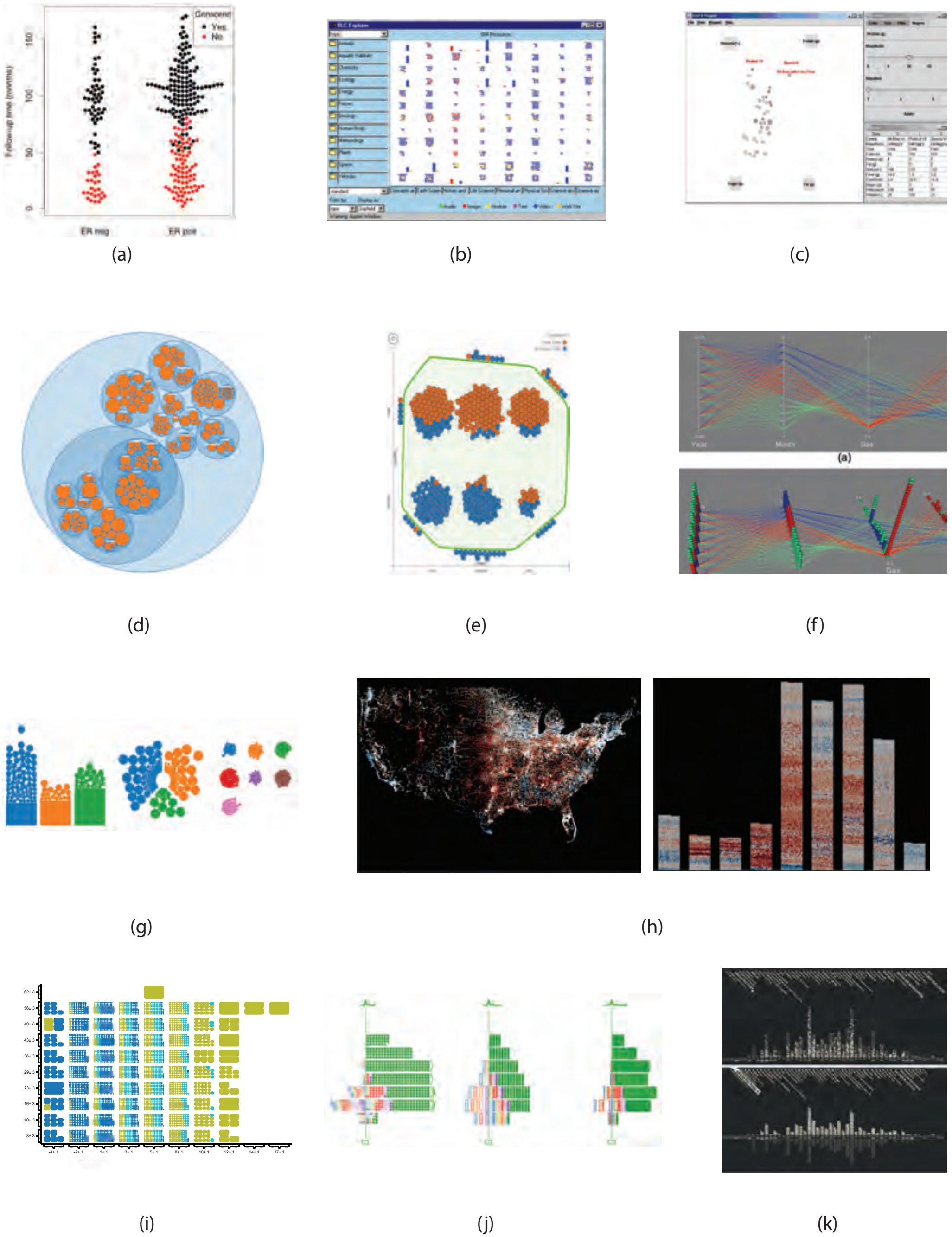
Unit Visualization

	Group	Income
Bill Gates	A	60
Steve Jobs	A	30
John Doe	B	2
…	…	…



Long tradition

Not new idea.



Why Unit Visualization?

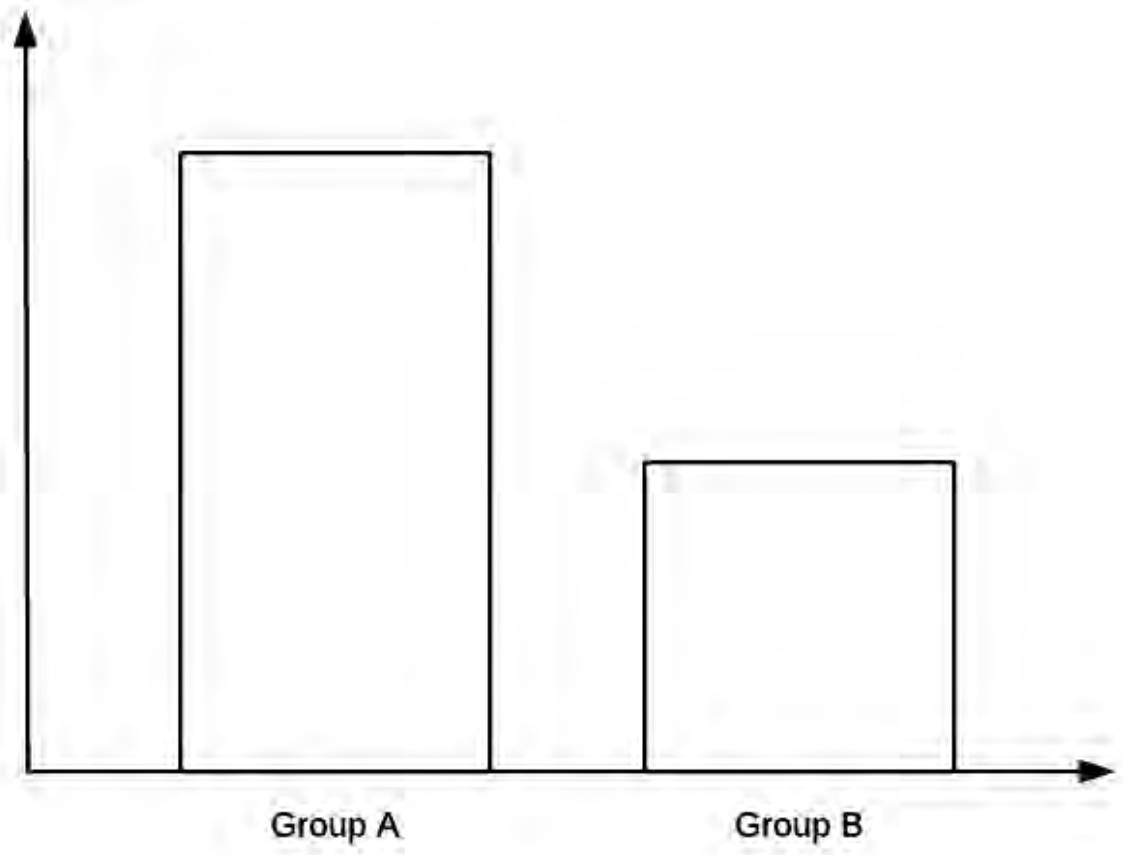
Benefit of Unit Visualizations Process

- Delivers More information
- Provides Natural format for Perception
- Enables Physical interactions

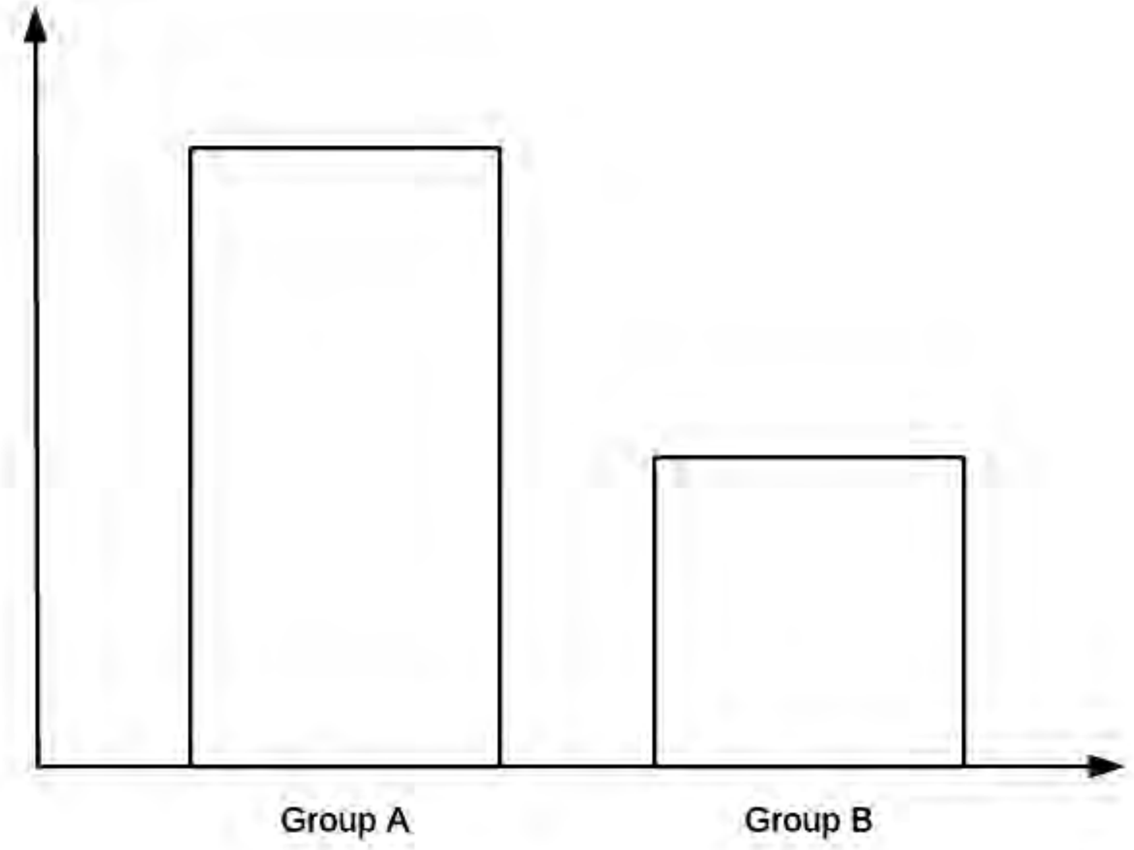
Benefit of Unit Visualizations Process

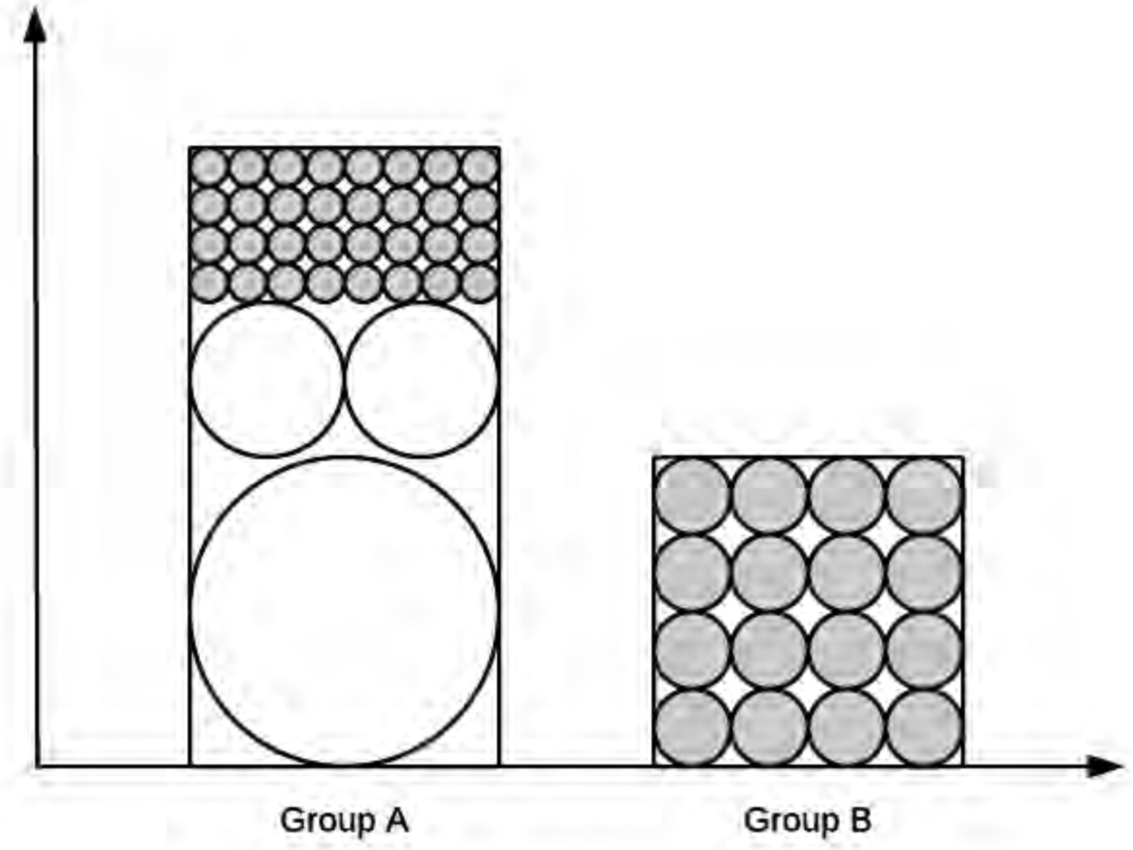
- Delivers More information
- Provides Natural format for Perception
- Enables Physical interactions

Total Income of Two Group

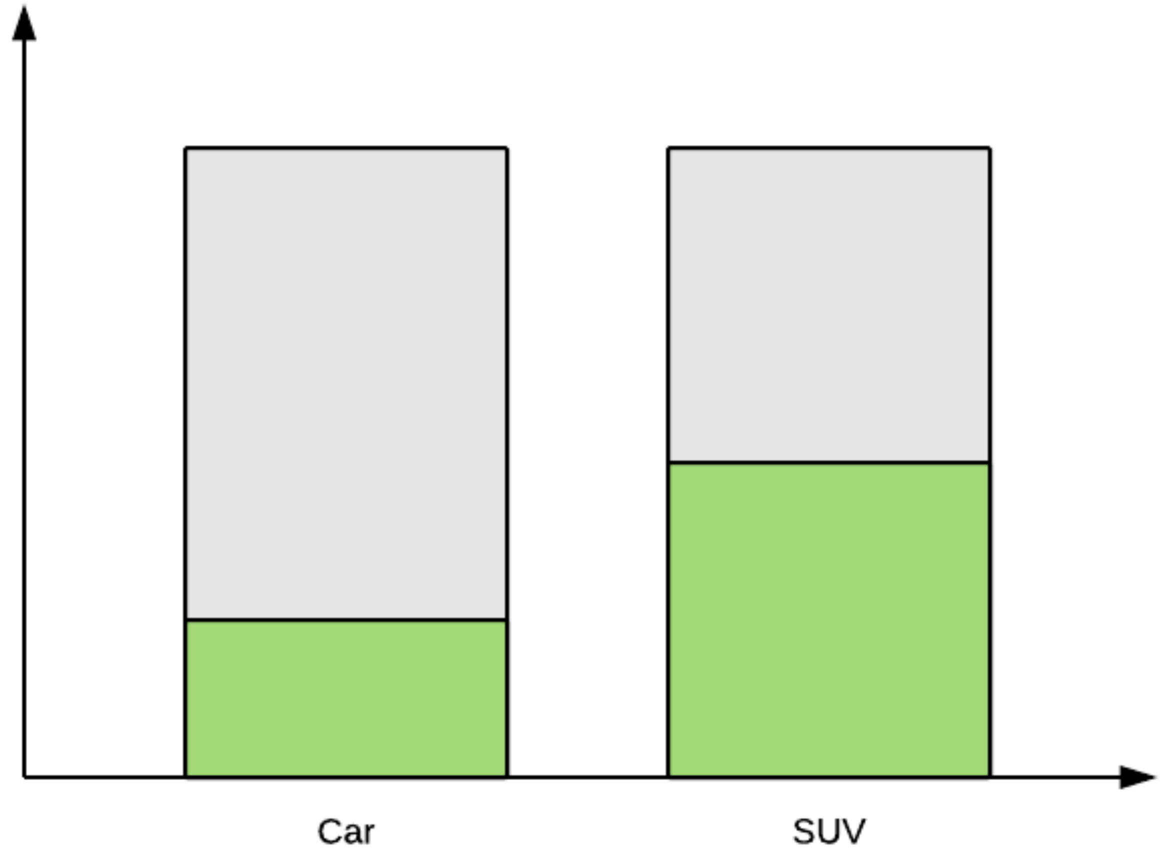


Total Income of Two Group

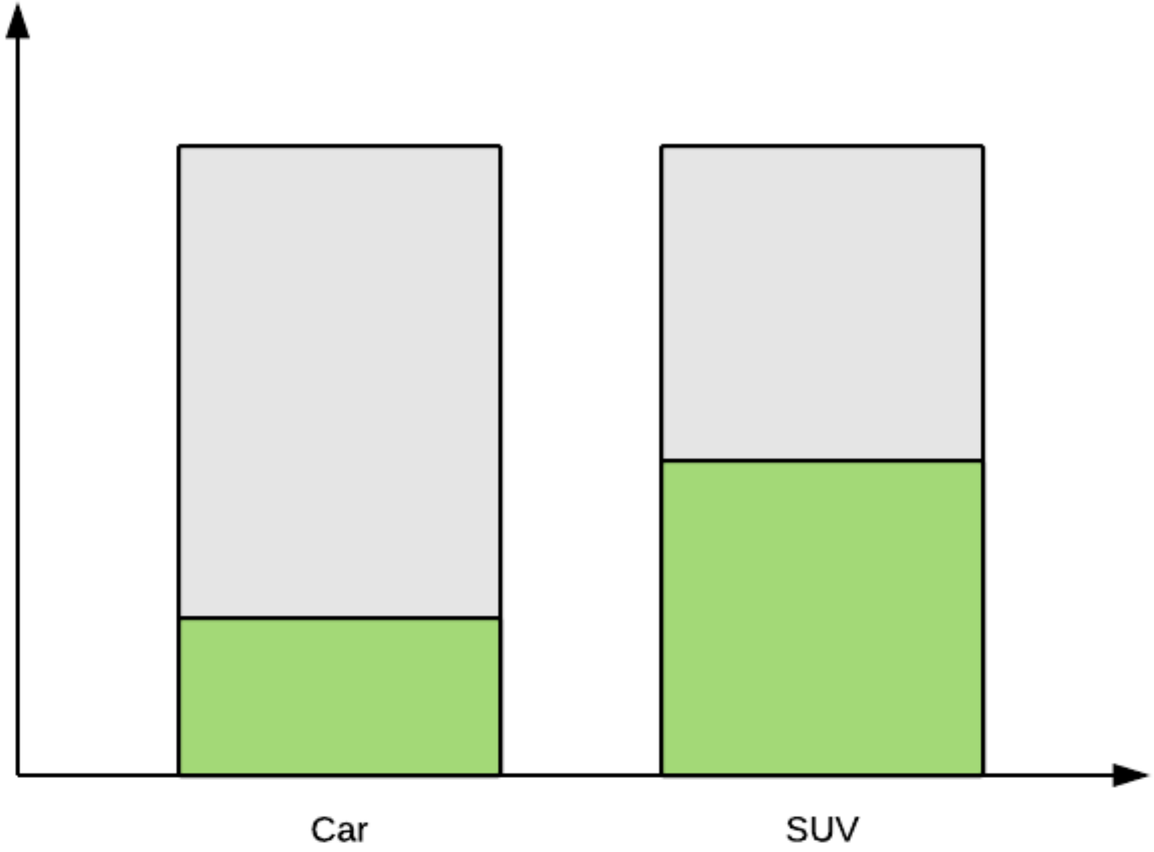


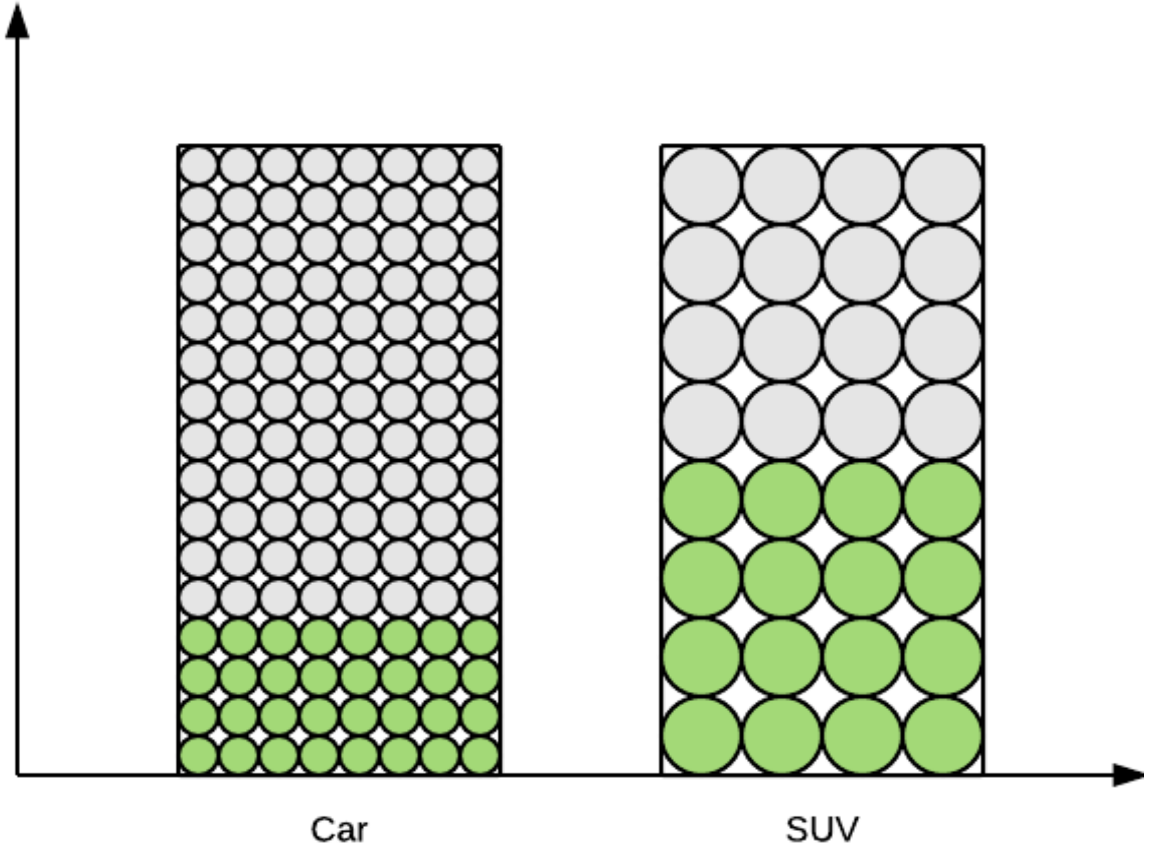


Claim by Car Type

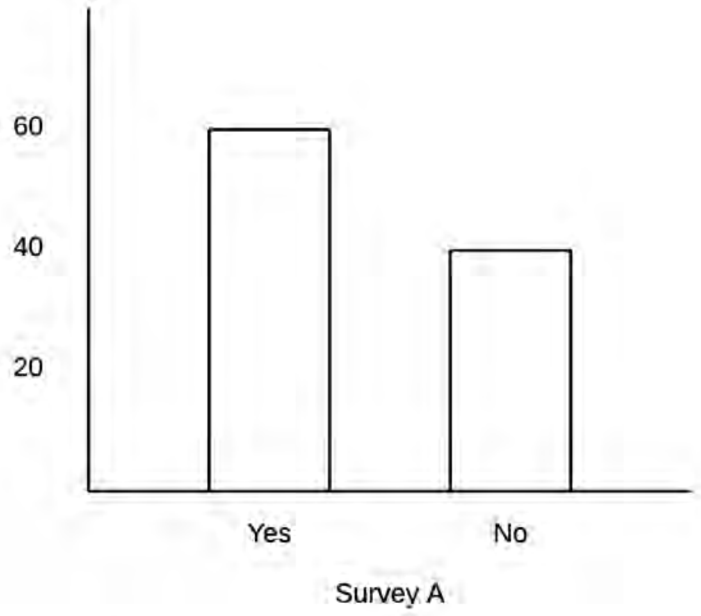


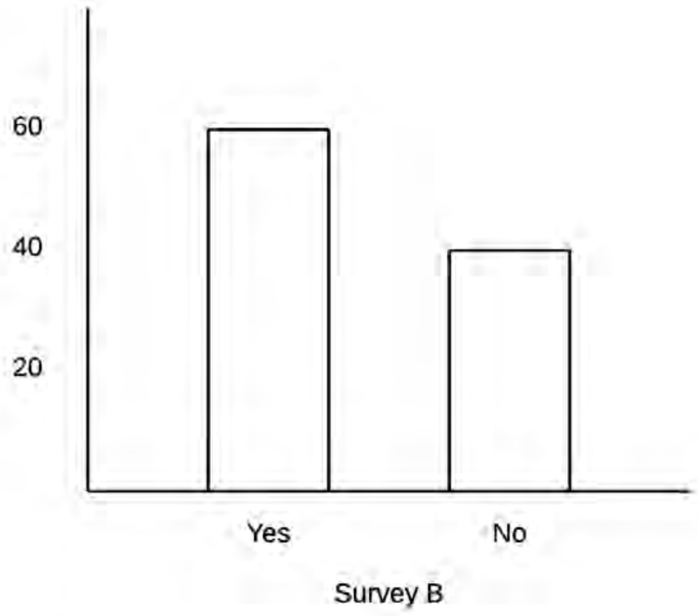
Claim by Car Type



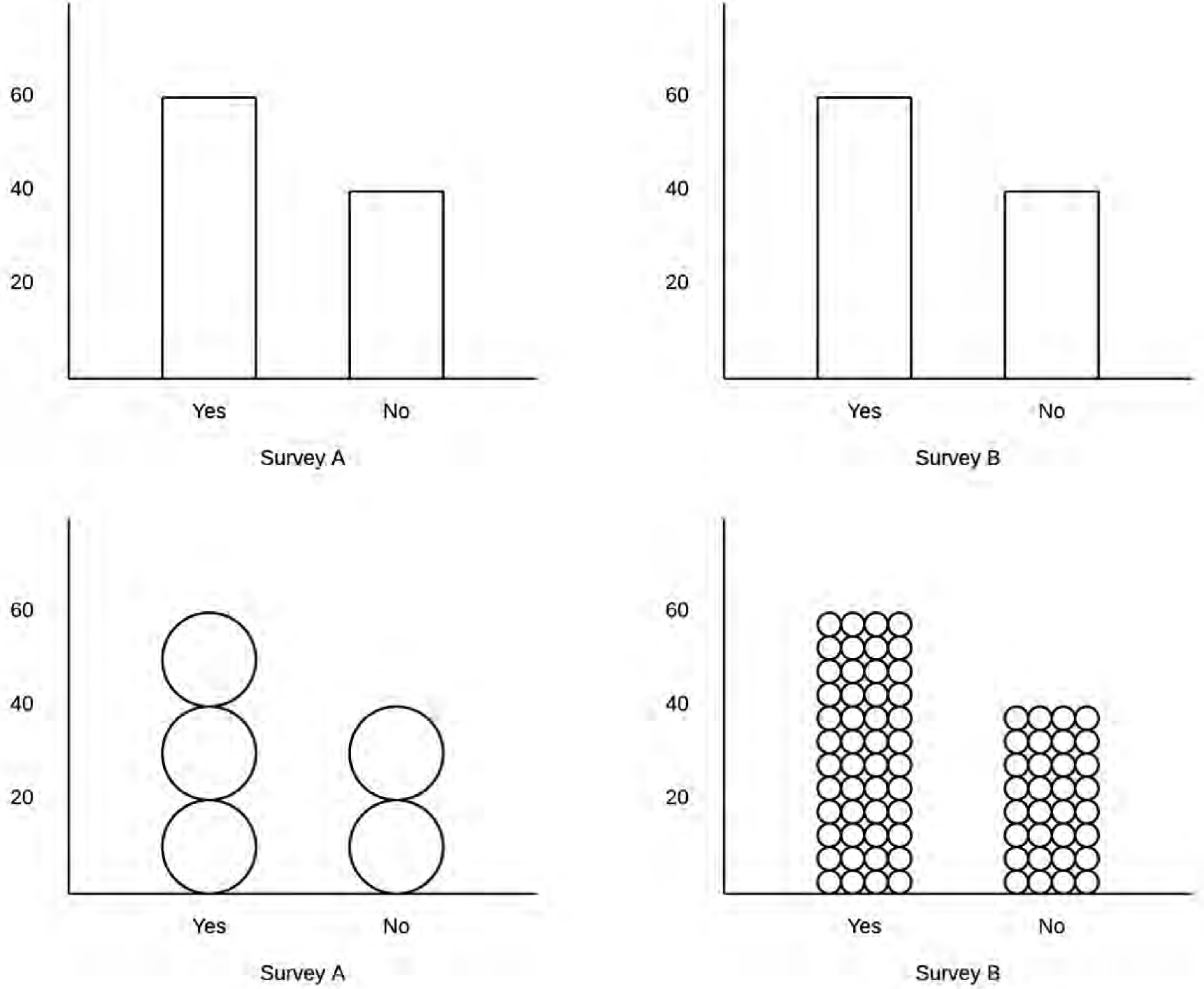


Survey Result

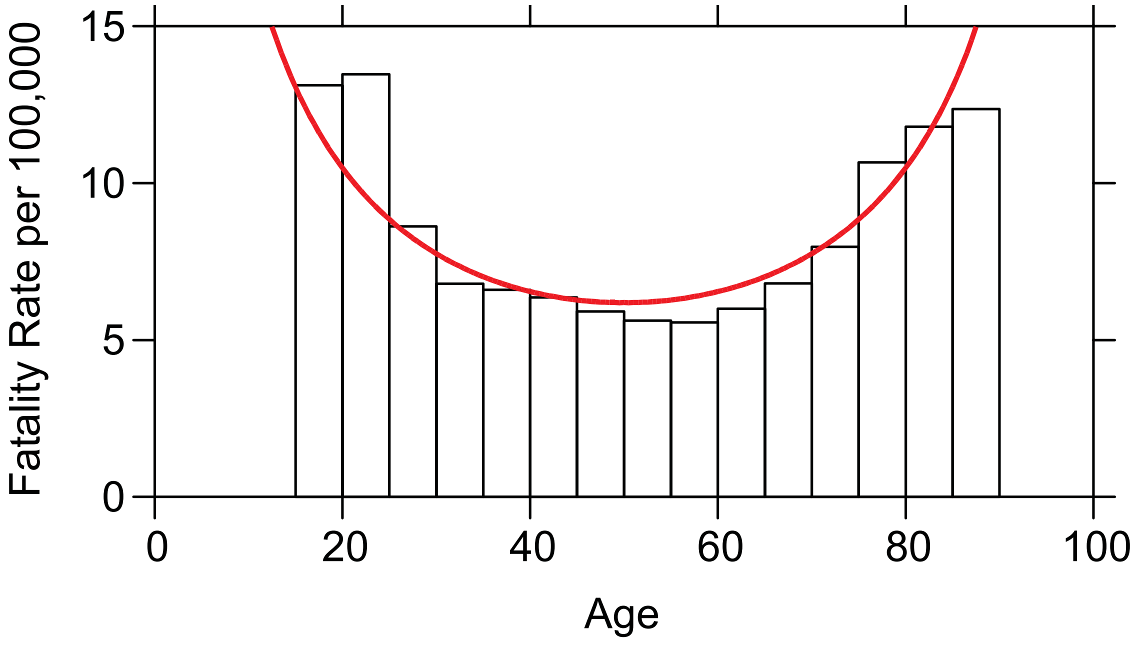




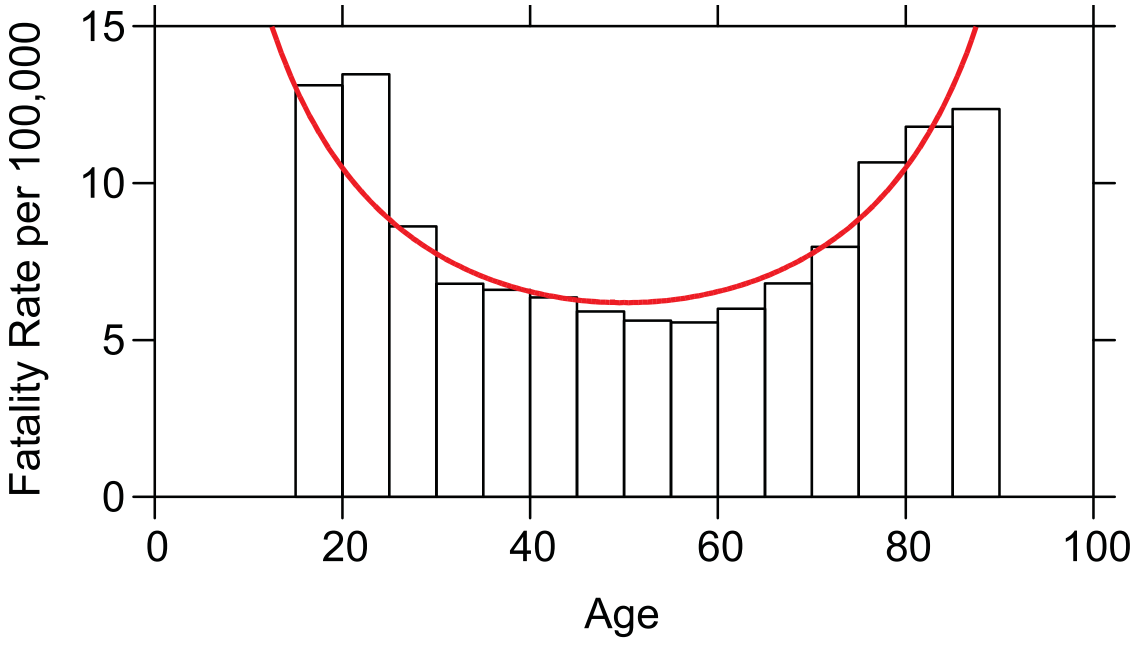
Survey Result



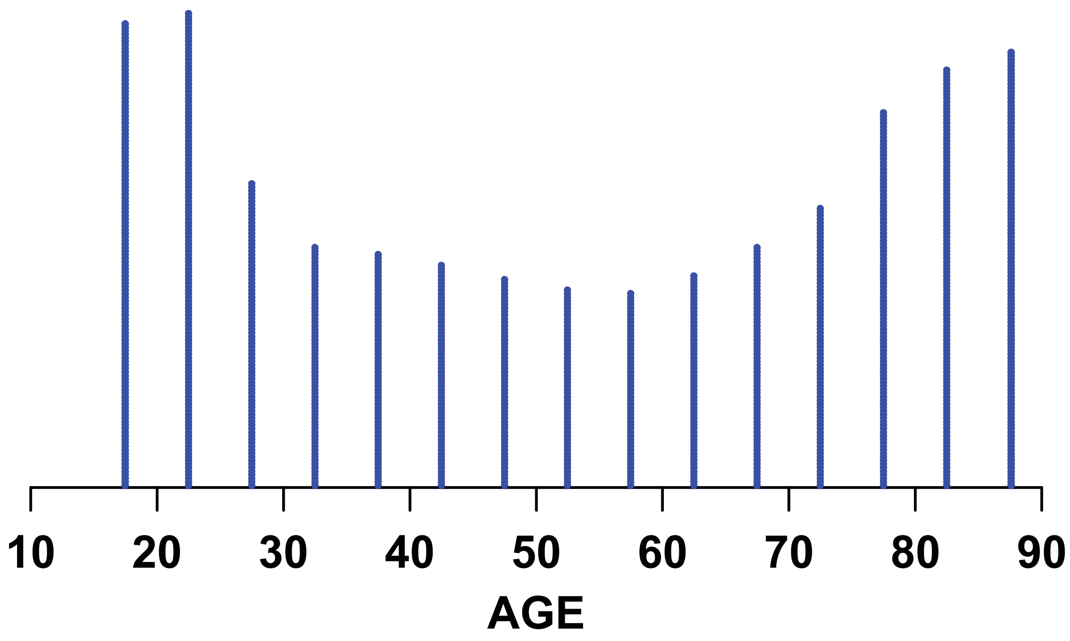
Traffic fatality rates by age



Traffic fatality rates by age



Histogram



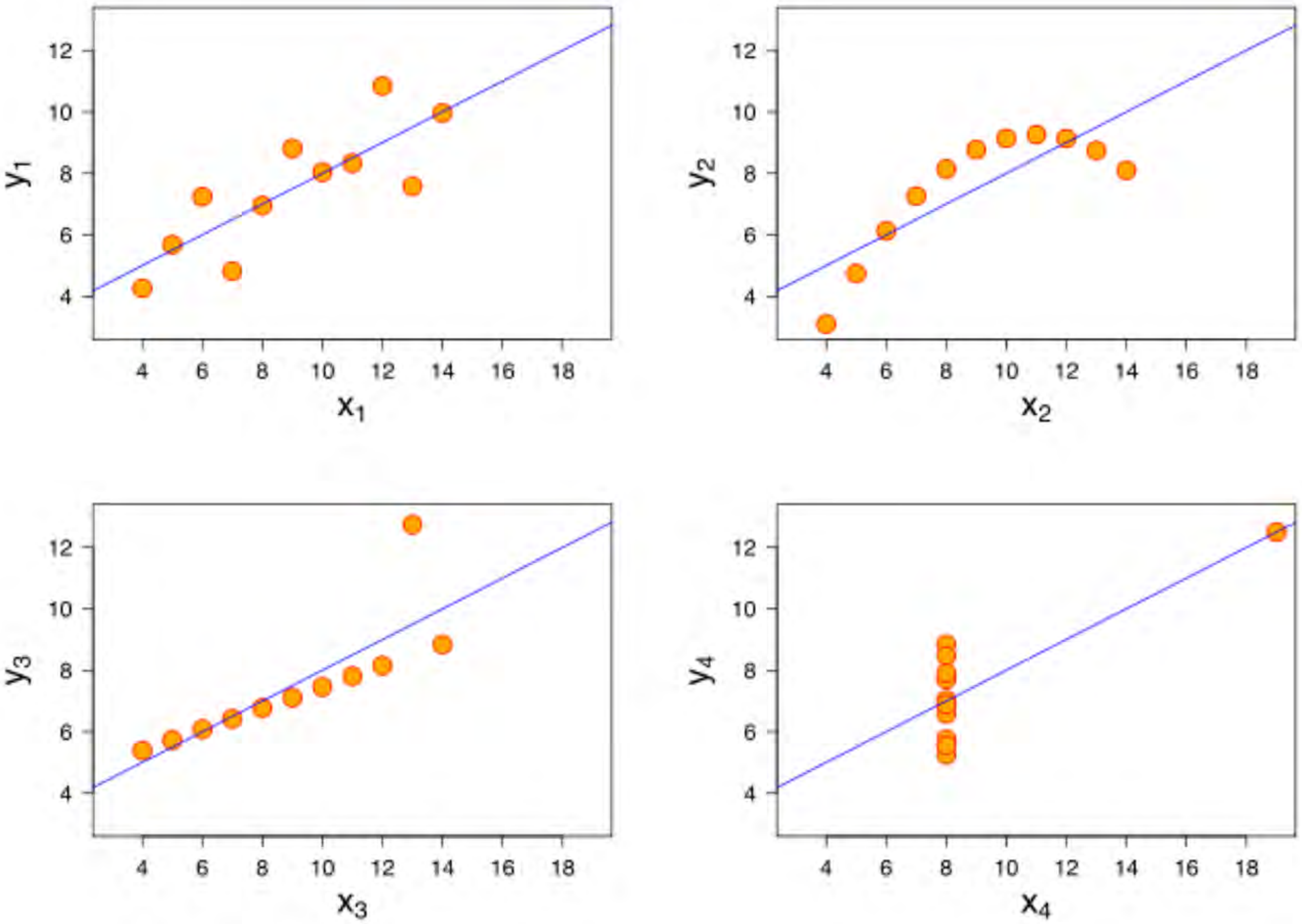
Dot Plots

Unexpected discovery was possible because it contained more information than user originally asked.

Anscombe's Quartet

I		II		III		IV	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

Property	Value
Mean of x in each case	9 (exact)
Sample variance of x in each case	11 (exact)
Mean of y in each case	7.50 (to 2 decimal places)
Sample variance of y	4.122 or 4.127 (to 3 decimal places)
Correlation between x and y	0.816 (to 3 decimal places)
Linear regression	$y = 3.00 + 0.500x$ (to 2 and 3 decimal places, respectively)



12
10
8
6
4
4 6 8 10 12 14 16 18
y_1
x_1
y_2
x_2
y_3
x_3
y_4
x_4

Aggregation =

Loss of information

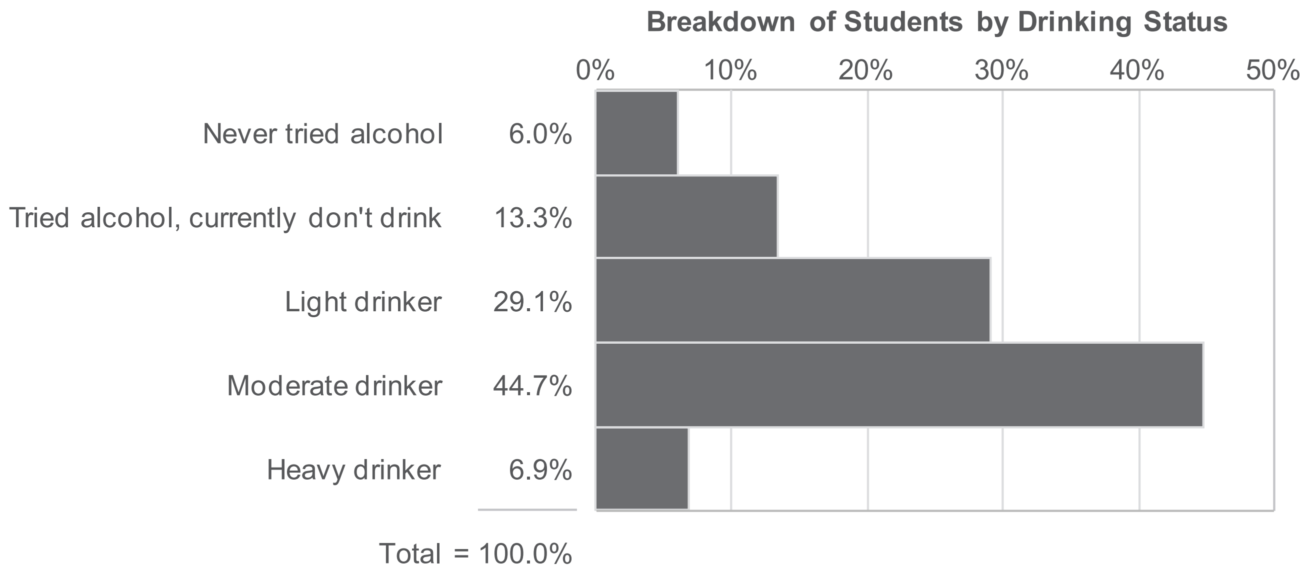
Benefit of Unit Visualizations Process

- Delivers More information
- Provides Natural format for Perception
- Enables Physical interactions

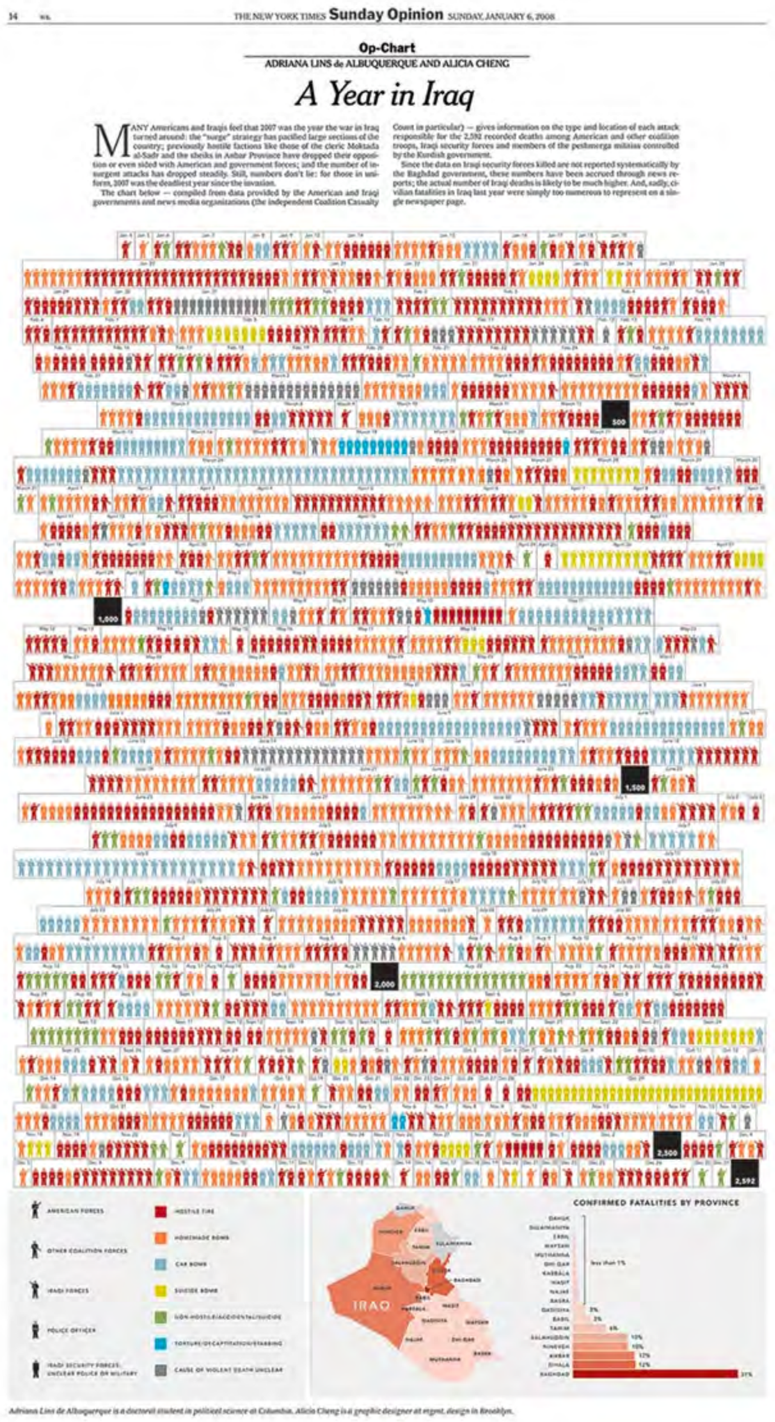
Benefit of Unit Visualizations Process

- Deliver More information
- Provides Natural format for Perception
- Enables Physical interactions

Source: E. J. Fox, from a larger infographic titled “#1 Party School”, based on data from “The Partnership Campus & Community United Against Dangerous Drinking Annual Assessment Report 08-09”



From “Unit Charts are for kids” by Stephen Few



THE NEW YORK TIMES **Sunday Opinion** SUNDAY, JANUARY 6, 2008

Op-Chart

ADRIANA LINS de ALBUQUERQUE AND ALICIA CHENG

A Year in Iraq

CONFIRMED FATALITIES BY PROVINCE

Still Journalism loves it.

Why?

Bayesian Inference Problem

The probability of breast cancer is 1% for a woman at 40 who participates in a routine screening. If a woman has breast cancer, the probability is 80% that she will have a positive mammography. If a woman does not have breast cancer, the probability is 9.6% that she will also have a positive mammography.

A woman in this age group had a positive mammography in a routine screening. What is the probability that she actually has breast cancer?

A. greater than 90%
B. between 70% and 90% — 95 out of 100 doctors
C. between 50% and 70%
D. between 30% and 50%
E. between 10% and 30%
F. less than 10% — Correct answer

"Base rate neglect"

From "Bayesian models of human learning and inference" by Josh Tenenbaum

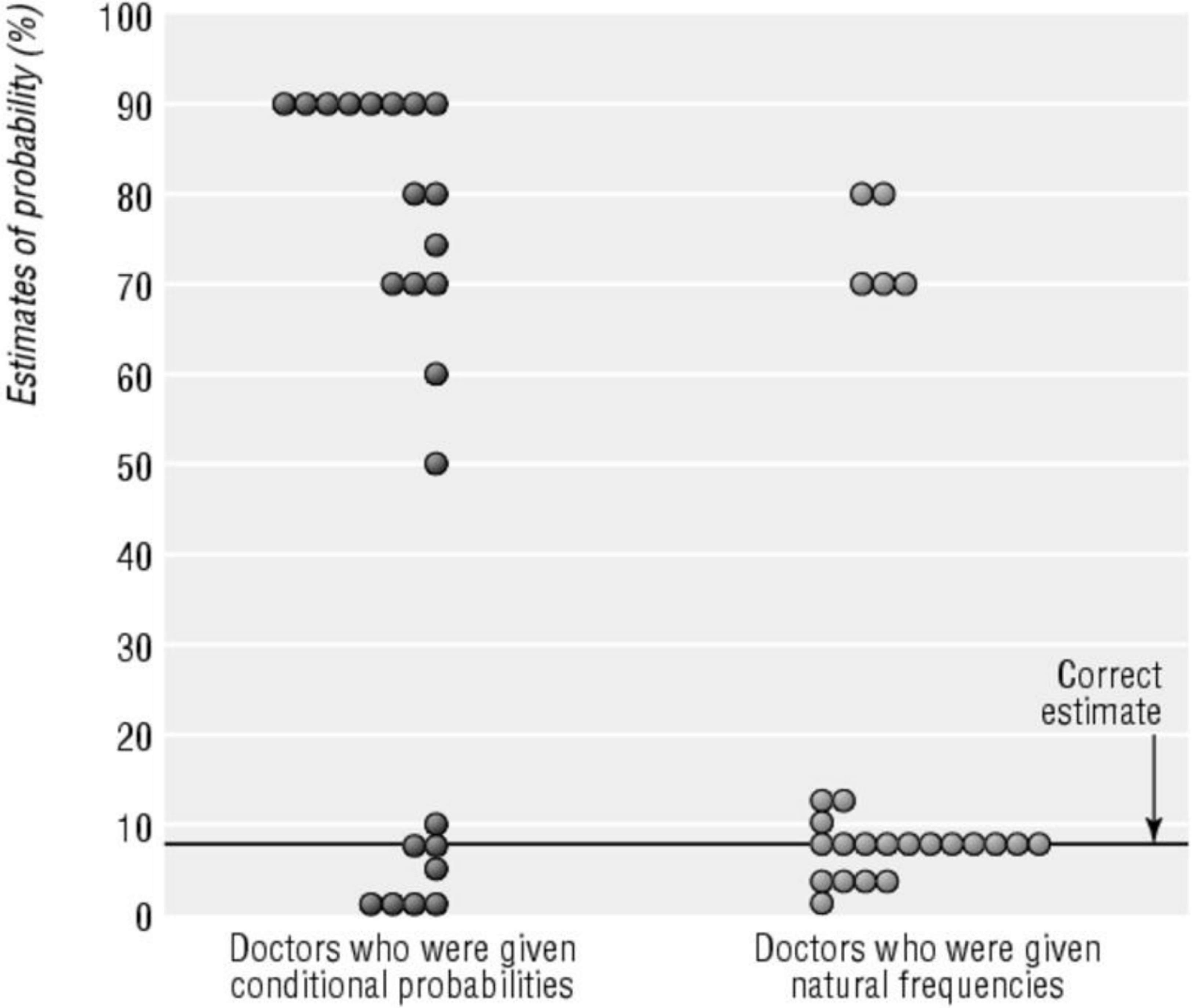
People aren’t Bayesian

- Kahneman and Tversky (1970’s-present): “heuristics and biases” research program. 2002 Nobel Prize in Economics.
- Slovic, Fischhoff, and Lichtenstein (1976): “It appears that people lack the correct programs for many important judgmental tasks.... it may be argued that we have not had the opportunity to evolve an intellect capable of dealing conceptually with uncertainty.”
- Stephen Jay Gould (1992): “Our minds are not built (for whatever reason) to work by the rules of probability.”

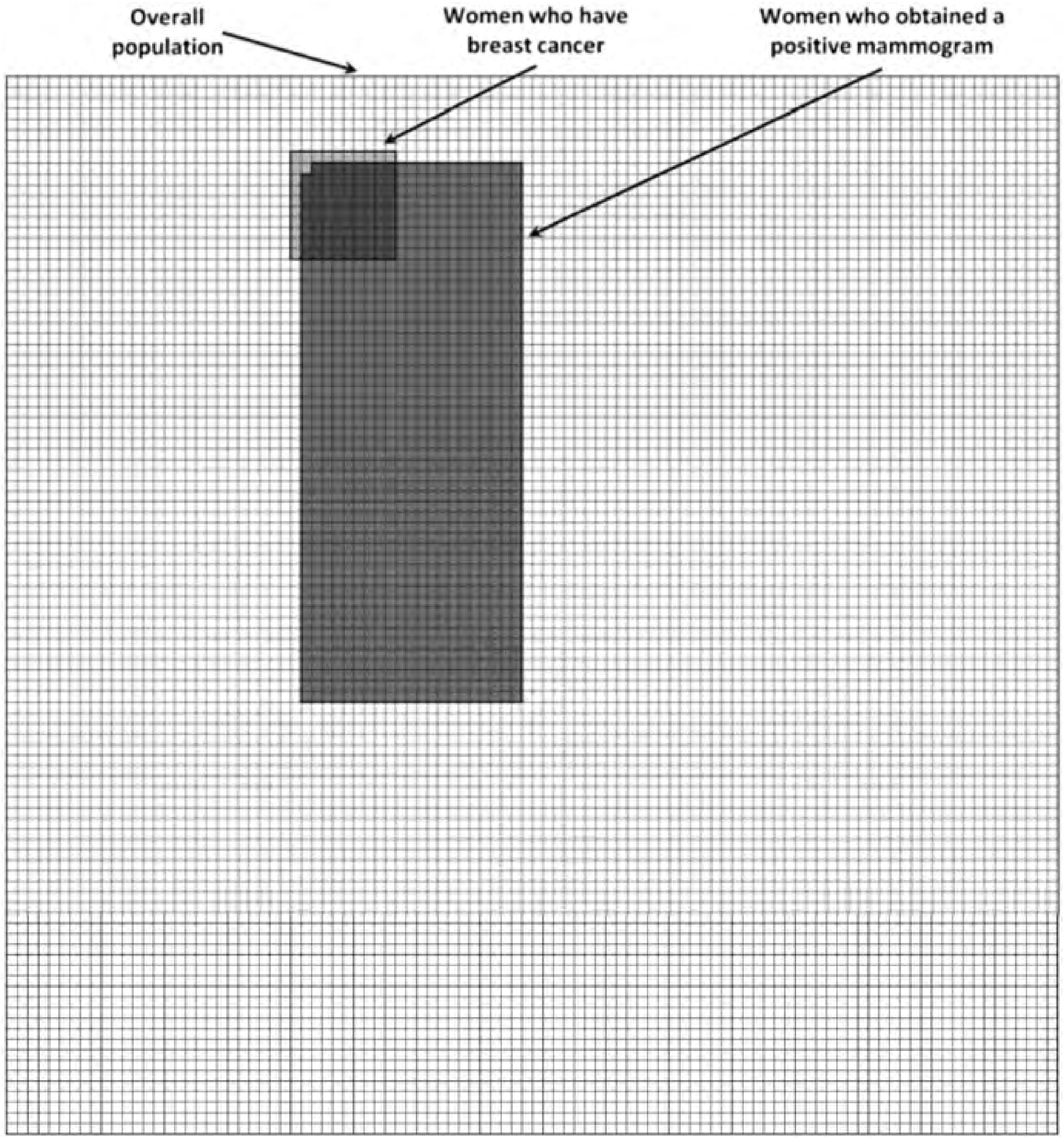
Environment of Evolutionary Adaptiveness (EEA)

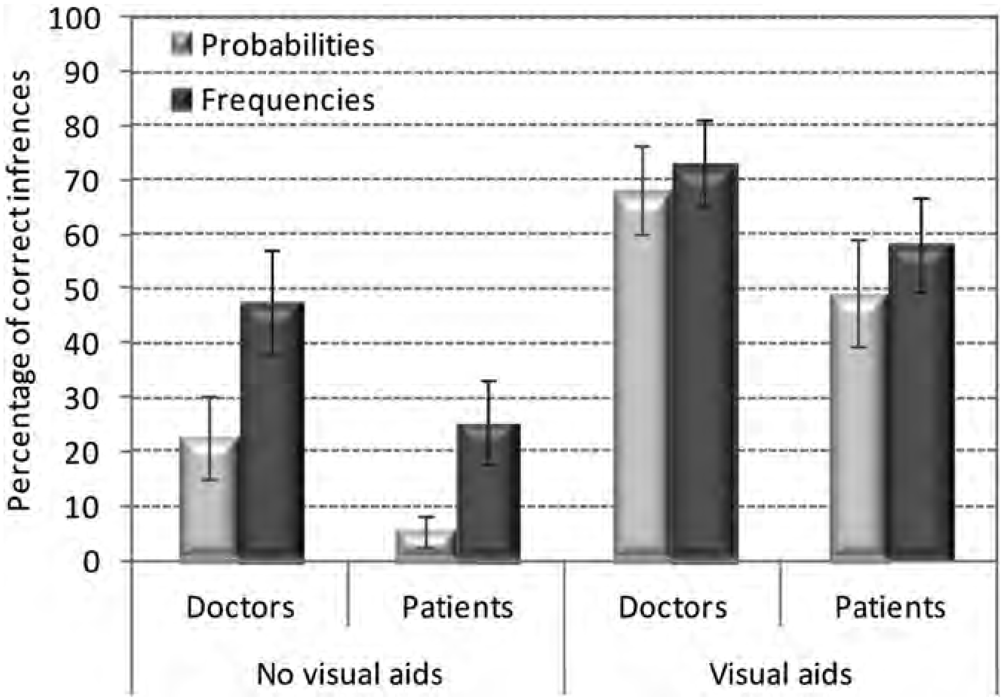
- **Conditional probabilities**
  - The probability that a woman has breast cancer is 0.8%. If she has breast cancer, the probability that a mammogram will show a positive result is 90%. If a woman does not have breast cancer the probability of a positive result is 7%. Take, for example, a woman who has a positive result. What is the probability that she actually has breast cancer?

- **Natural frequencies**
  - Eight out of every 1000 women have breast cancer. Of these eight women with breast cancer seven will have a positive result on mammography. Of the 992 women who do not have breast cancer some 70 will still have a positive mammogram. Take, for example, a sample of women who have positive mammograms. How many of these women actually have breast cancer?



From “Simple tools for understanding risks: from innumeracy to insight” by Gerd Gigerenzer and Adrian Edwards





From "Visual representation of statistical information improves diagnostic inferences in doctors and their patients" by Garcia-Retamero and Hoffrage

Assessing the Effect of Visualizations on Bayesian Reasoning through Crowdsourcing

Luana Micallef, Pierre Dragicevic, and Jean-Daniel Fekete, *Member, IEEE*

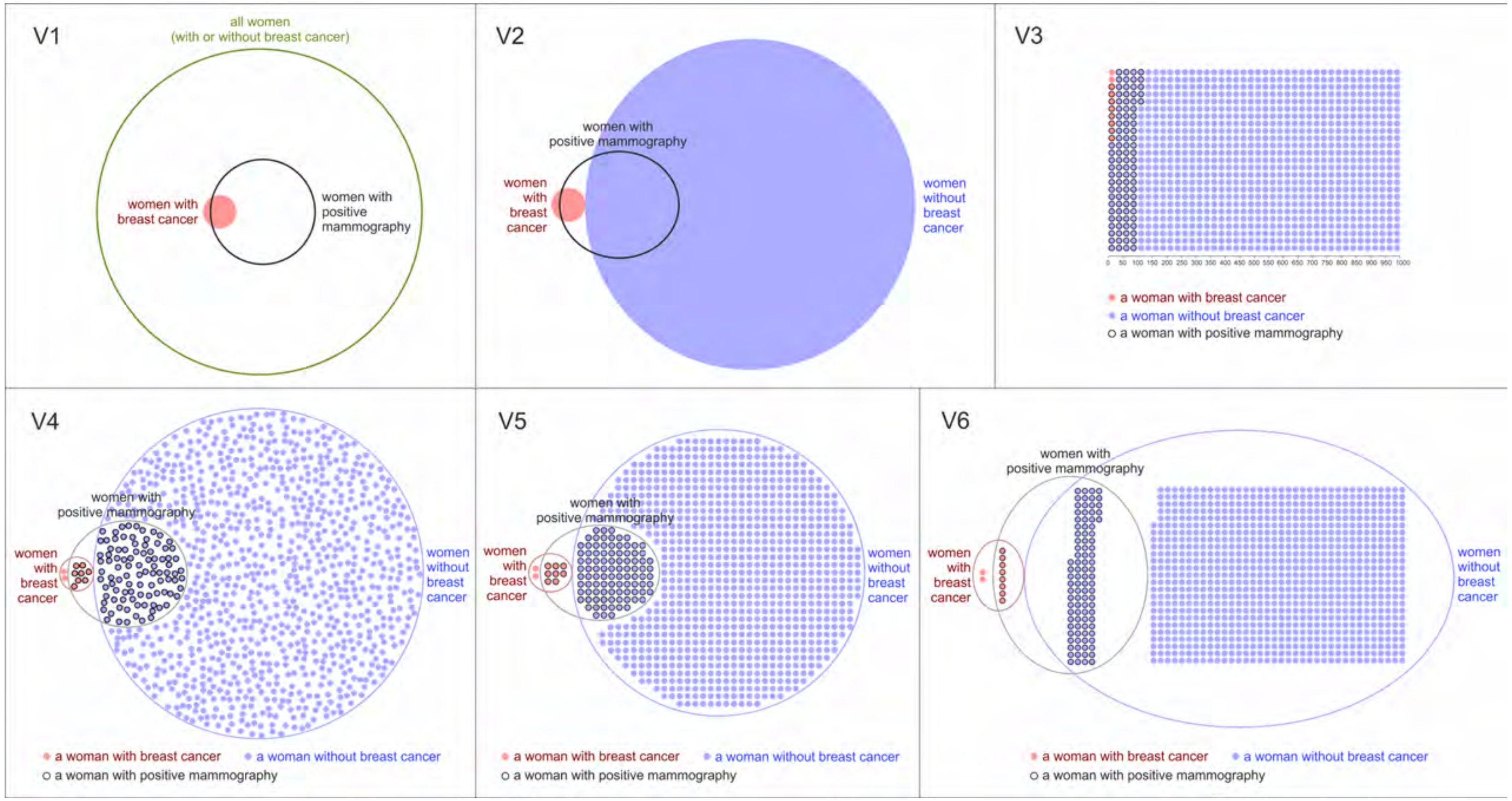
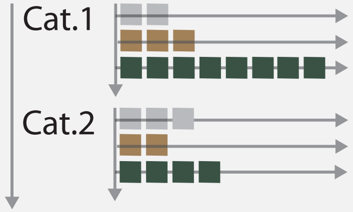
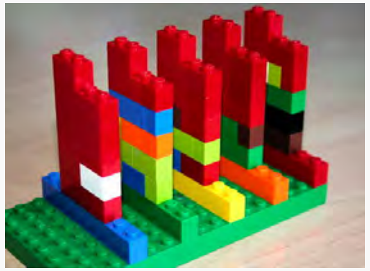
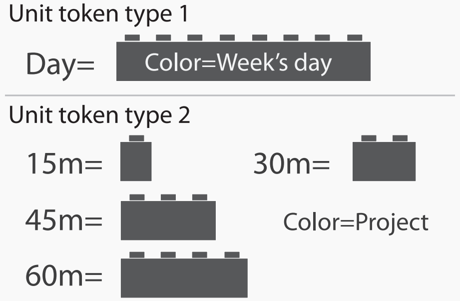
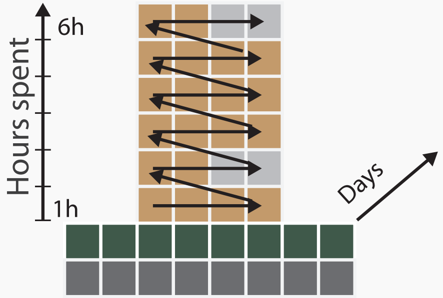
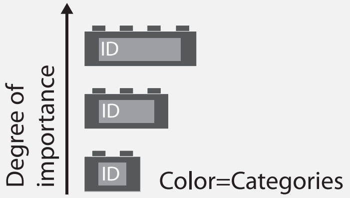
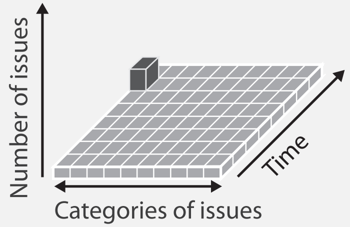


Fig. 1. The six visualizations evaluated in our study, illustrating the classic mammography problem [21].

Examples	Picture fragment	Type	C1: Token	C2: Token Grammar	C4: Assembly model	C3: Environment
1. Chris Jordan		Artistic	Object / picture of object	1 picture: = 1 plastic cup, used on airline flights in the US during last six hours	Artistic, the assembly model in this case does not follow the definition of a monosemic system. The assembly is not described as processing the data, but as providing a feeling about the data.	2D Paper canvas
2. Otto Neurath		Analytic	Pictogram	3 Pictograms: = 1 car per 50 people, = 1 bus per 50 people, = 1 phone per 50 people		2D Paper canvas
3. Michael Hunger		Analytic	Lego bricks			3D Physical tangible Lego board
4. Kevin Quinn		Analytic	Lego bricks			3d Physical tangible Lego board

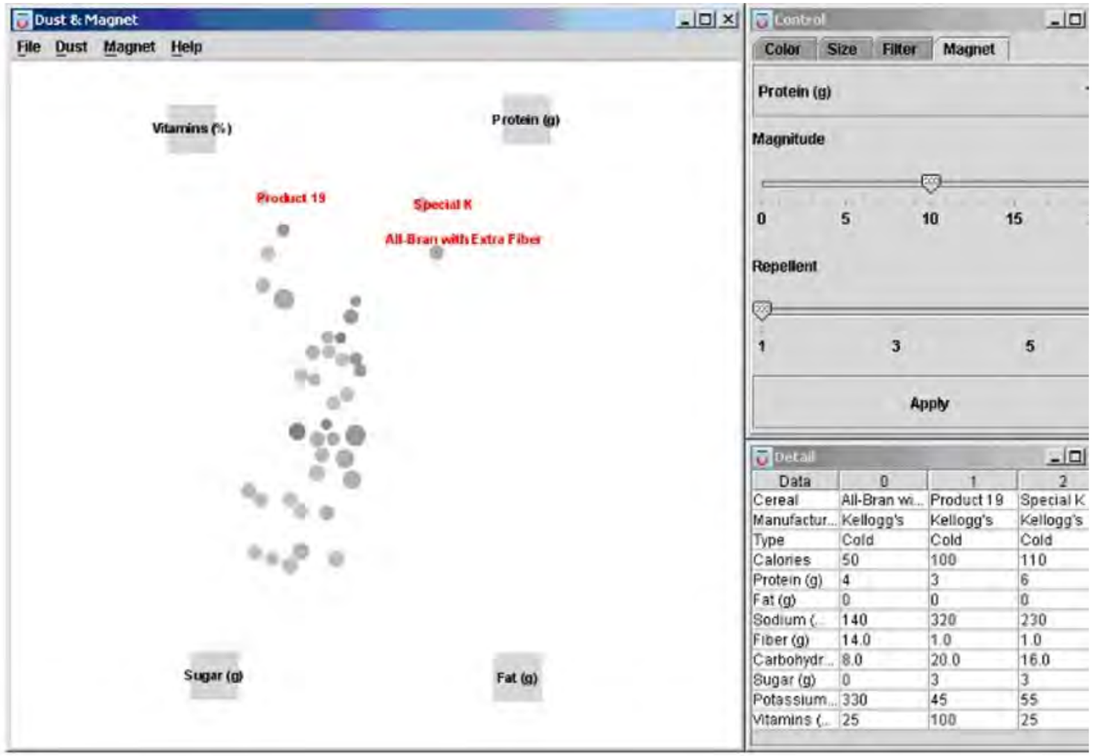
"Constructive Visualization" by Huron et al

Benefit of Unit Visualizations Process

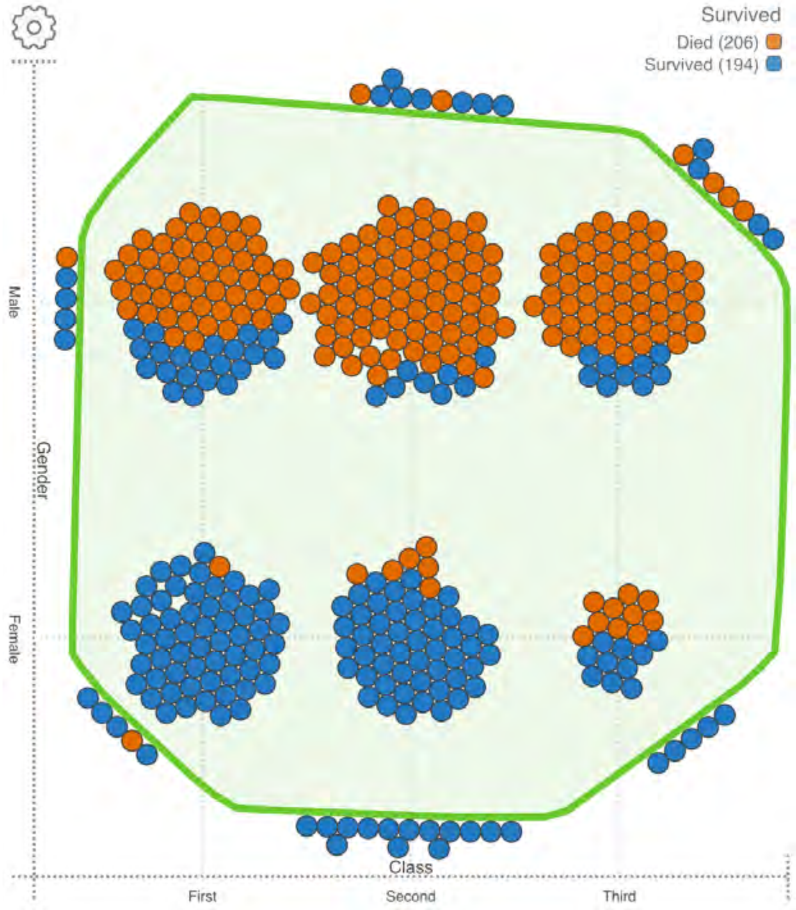
- Deliver More information
- Provides Natural format for Perception
- Enables Physical interactions

Benefit of Unit Visualizations Process

- Deliver More information
- Provides Natural format for Perception
- Enables Physical interactions



Dust and Magnet by Yi et al.



Kinetica: Naturalistic Multi-touch Data Visualization by Rzeszotarski and Kittur

Constructing Visual Representations: Investigating the Use of Tangible Tokens

Samuel Huron, Yvonne Jansen, Sheelagh Carpendale

Fig. 1. Constructing a visualization with tokens: right hand positions tokens, left hand points to the corresponding data.

Abstract—The accessibility of infovis authoring tools to a wide audience has been identified as a major research challenge. A key task in the authoring process is the development of visual mappings. While the infovis community has long been deeply interested in finding *effective* visual mappings, comparatively little attention has been placed on *how* people construct visual mappings. In this paper, we present the results of a study designed to shed light on how people transform data into visual representations. We asked people to create, update and explain their own information visualizations using only tangible building blocks. We learned that all participants, most of whom had little experience in visualization authoring, were readily able to create and talk about their own visualizations. Based on our observations, we discuss participants' actions during the development of their visual representations and

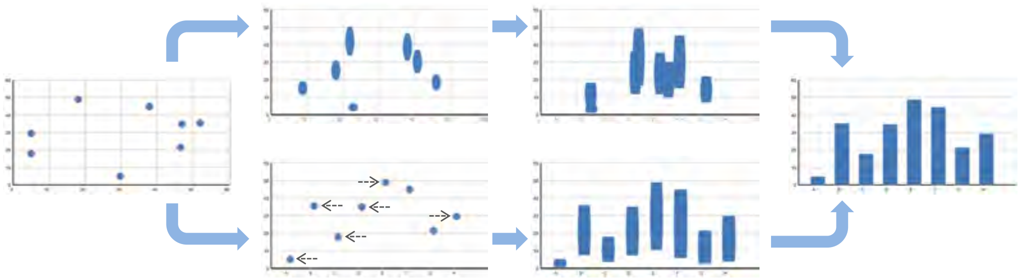


Figure 1. Animating from a scatter plot to a bar chart. The top path directly interpolates between the starting and ending states. The bottom path is staged: the first stage moves points to their x-coordinates and updates the x-axis, the second stage morphs the points into bars.

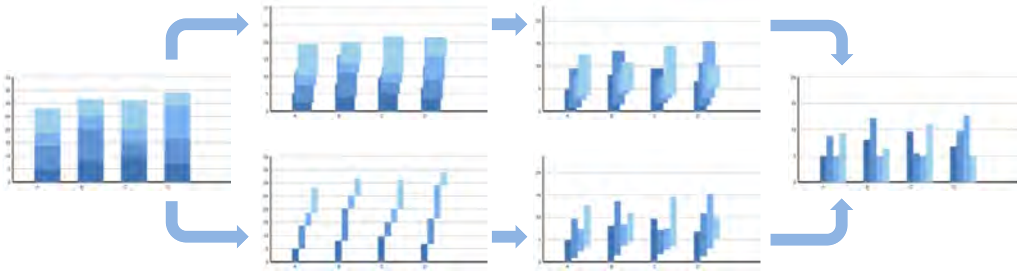


Figure 2. Animating from stacked bars to grouped bars. The top path directly interpolates between the starting and ending states. The bottom path is staged: the first stage changes the widths and x-coordinates of bars, the second stage drops the bars down to the baseline.

Figure 3. A multi-stage animation of changing values in a donut chart. Stage 1: Wedges split into two rings. Stage 2: Wedges translate to be centered on their final position. Stage 3: Wedges then update their values, changing size. Stage 4: Wedges reunite into a single ring.

Animated Transitions in Statistical Data Graphics
By Jeffrey Heer, George G. Robertson

Visual Sedimentation

Samuel Huron, Romain Vuillemot, and Jean-Daniel Fekete, *Senior Member, IEEE*

Fig. 1. The Visual Sedimentation metaphor applied to a bar chart (left), a pie chart (center), and a bubble chart (right).

Abstract—We introduce *Visual Sedimentation*, a novel design metaphor for visualizing data streams directly inspired by the physical process of sedimentation. Visualizing data streams (*e. g.,* Tweets, RSS, Emails) is challenging as incoming data arrive at unpredictable rates and have to remain readable. For data streams, clearly expressing chronological order while avoiding clutter, and keeping aging data visible, are important. The metaphor is drawn from the real-world sedimentation processes: objects fall due to gravity, and aggregate into strata over time. Inspired by this metaphor, data is visually depicted as falling objects using a force model to land on a surface, aggregating into strata over time. In this paper, we discuss how this metaphor addresses the specific challenge of smoothing the transition between incoming and aging data. We describe the metaphor's design space, a toolkit developed to facilitate its implementation, and example applications to a range of case studies. We then explore the generative capabilities of the design space through our toolkit. We finally illustrate creative extensions of the metaphor when applied to real streams of data.

Index Terms—Design, information visualization, dynamic visualization, dynamic data, data stream, real time, metaphor

1 Introduction

This paper introduces *Visual Sedimentation*, a novel design metaphor inspired by the physical process of sedimentation. This process is the

nature, followed by an explanation of how it applies as a metaphor for data streams visualizations. To facilitate the design of visualizations

Benefit of Unit Visualizations Process

- Deliver More information
- Provides Natural format for Perception
- Enables Physical interactions

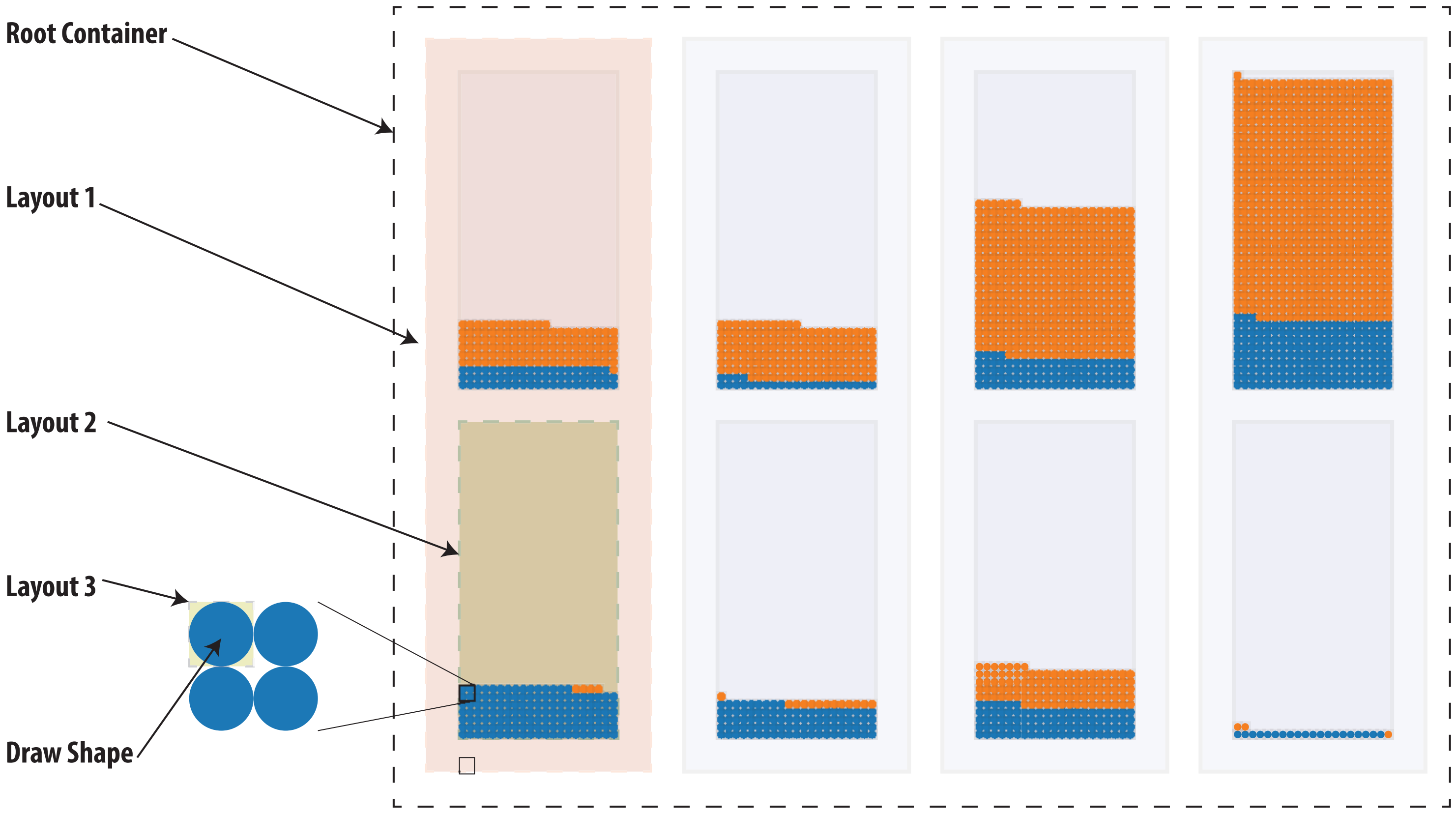
Benefit of Unit Visualizations Process

- Deliver More information
- Provides Natural format for Perception
- Enables Physical interactions

How?

ATOM: Unit Visualization Grammar

```
{
 "data": "data/titanic.csv",
 "width": 320, "height": 240,  "padding": {..},
 "layouts": [{
  "name": "layout1",
  "type": "gridxy",
  "subgroup": {  "type": "groupby",  "key": "Class" },
  "aspect_ratio": "fillX",
  "size": { "type": "uniform",  "isShared": false },
  "direction": "LRBT",  "align": "LB",
  "margin": { ... },  "padding": {... }
 }, {
  "type": "gridxy",
  "subgroup": { "type": "groupby", "key": "Sex", "isShared": true },
  "aspect_ratio": "fillY",
  "size": { "type": "uniform", "isShared": true },
  }, {
  "subgroup": { "type": "flatten" },
  "aspect_ratio": "square",
  "size": { "type": "uniform", "isShared": true },
  "sort": { "key": "Survived"}
 }],
 "mark": {
  "shape": "circle",
  "color": { "key": "Survived", "type": "categorical" },
  "size": { "type": "max",  "isShared": false },
  "isColorScaleShared": true
 }
}
```

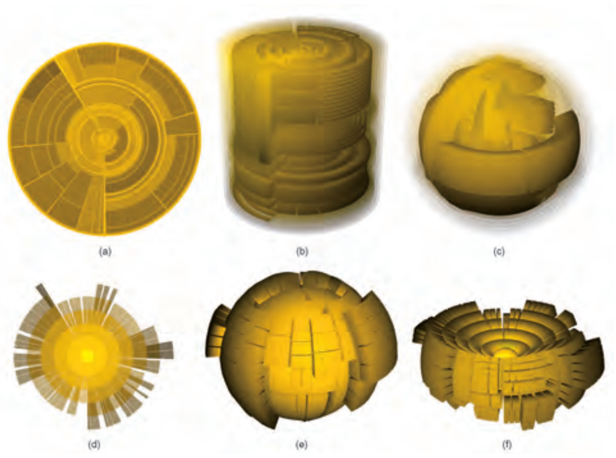


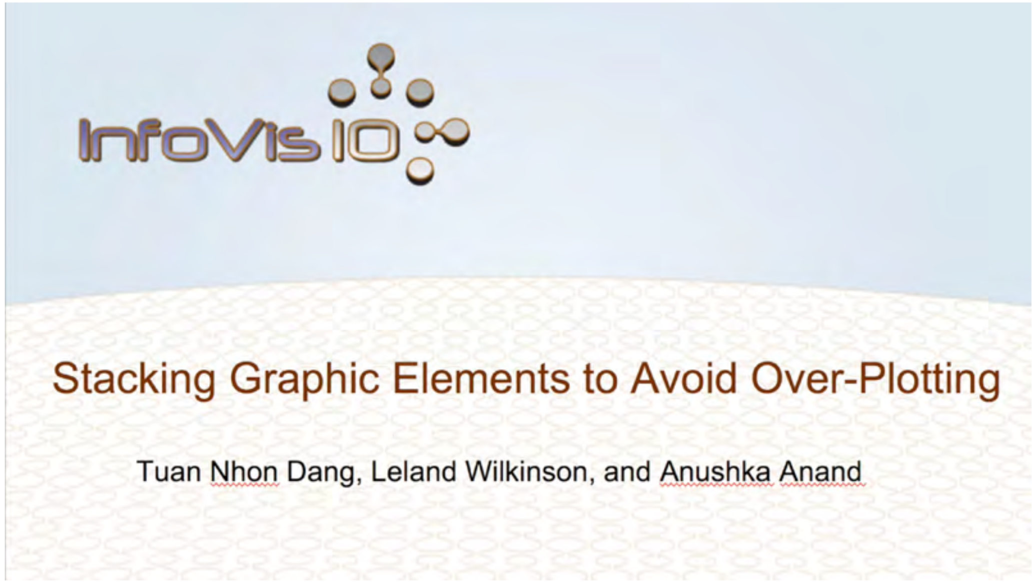
Design Space of Unit Visualizations

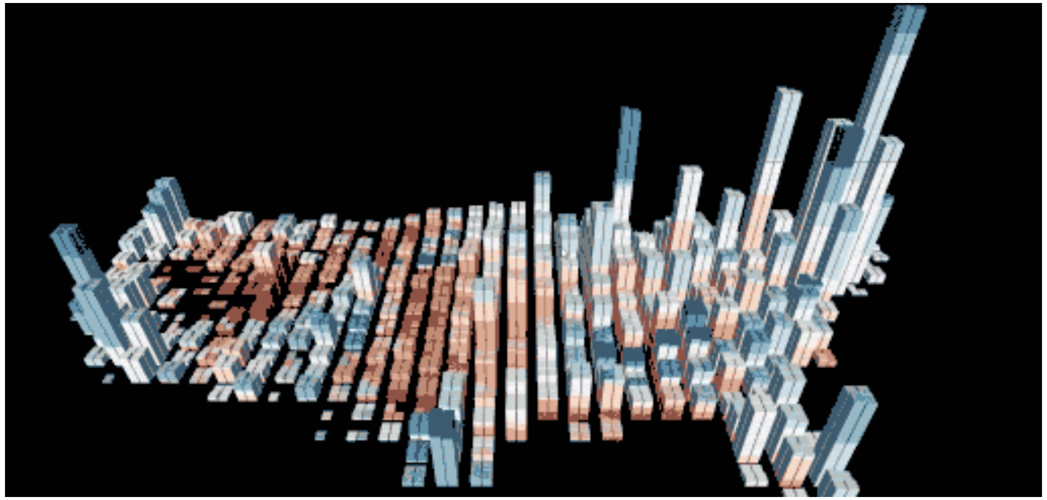
- Visual Space
- Layout
- Unit Representation

Visual Space

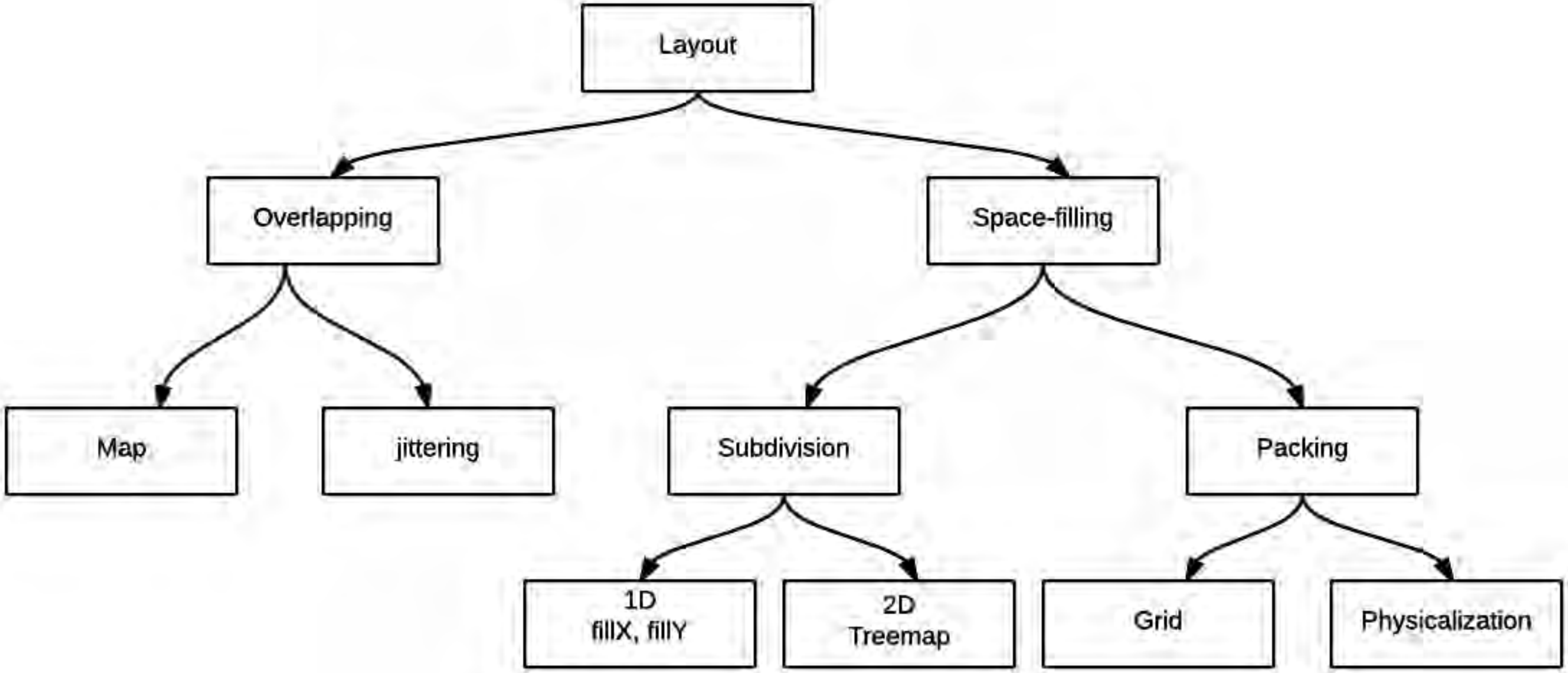
- Dimension
  - 1D, 2D, 3D
- Coord
  - Rect
  - Polar theta
  - Cylinder
  - Sphere
  - Map



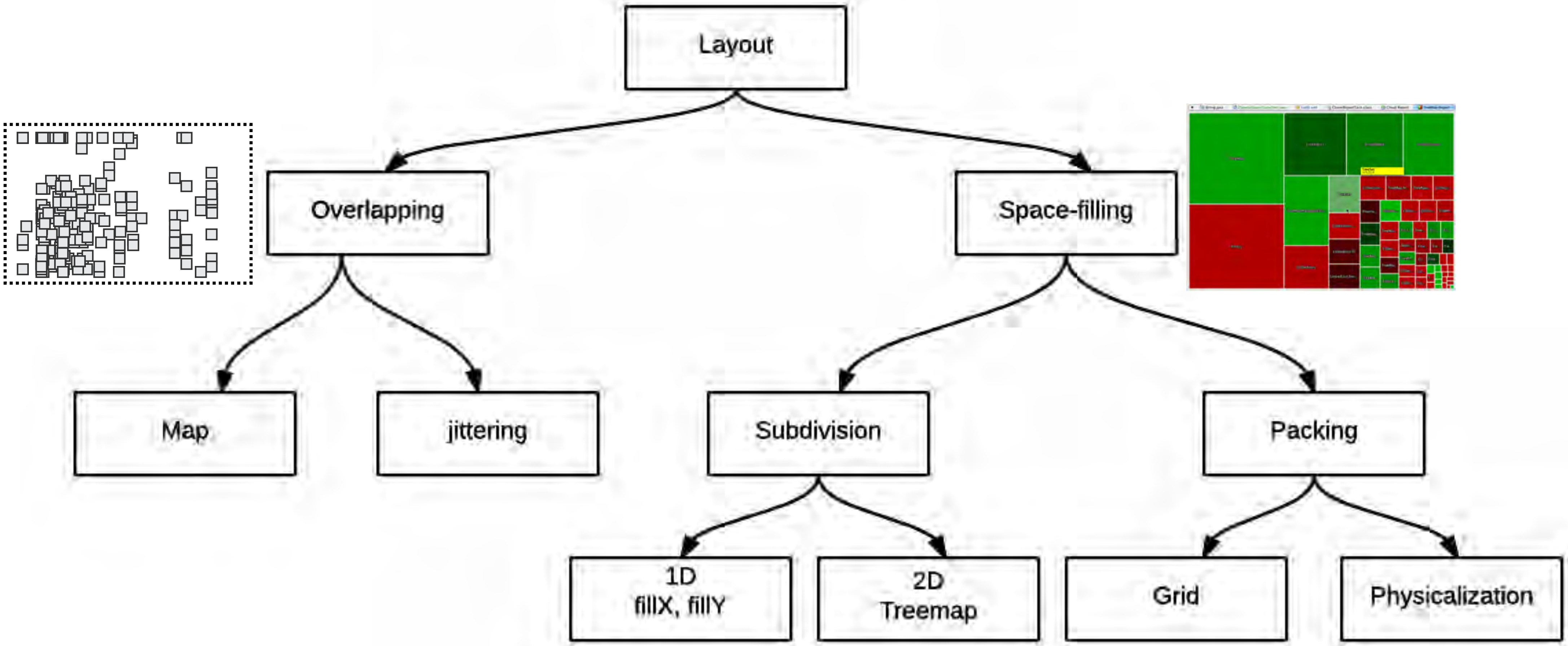




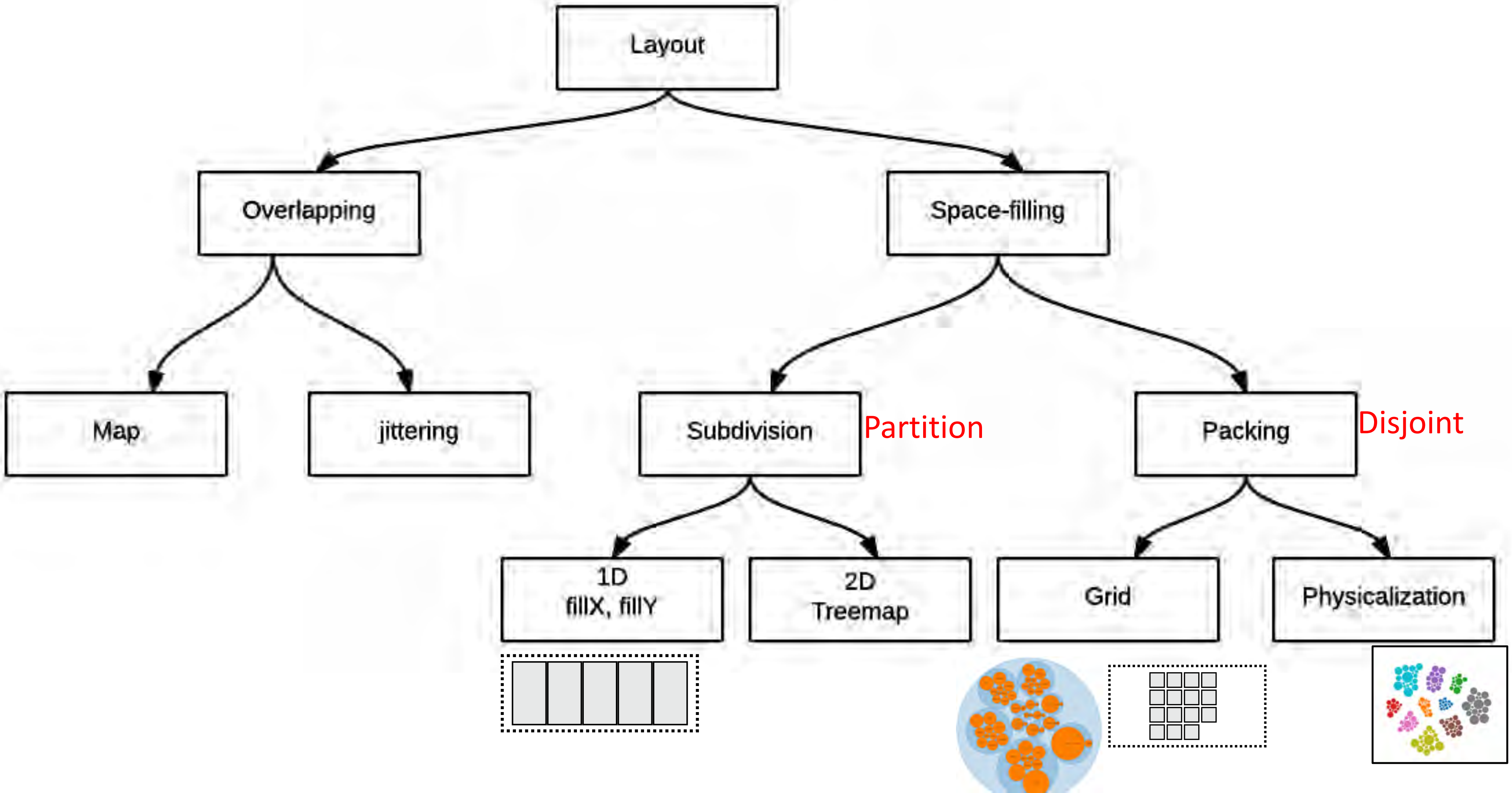
Layout


Layout
Overlapping
Space-filling
Map
jittering
Subdivision
Packing
1D
fillX, fillY
2D
Treemap
Grid
Physicalization

Layout

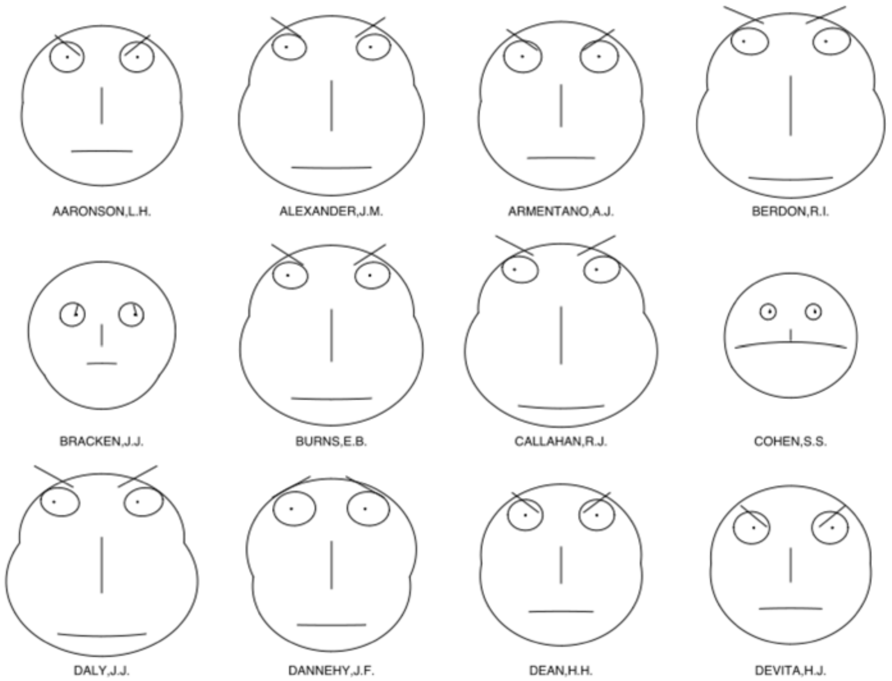

Layout
Overlapping
Space-filling
Map
jittering
Subdivision
Packing
1D
fillX, fillY
2D
Treemap
Grid
Physicalization

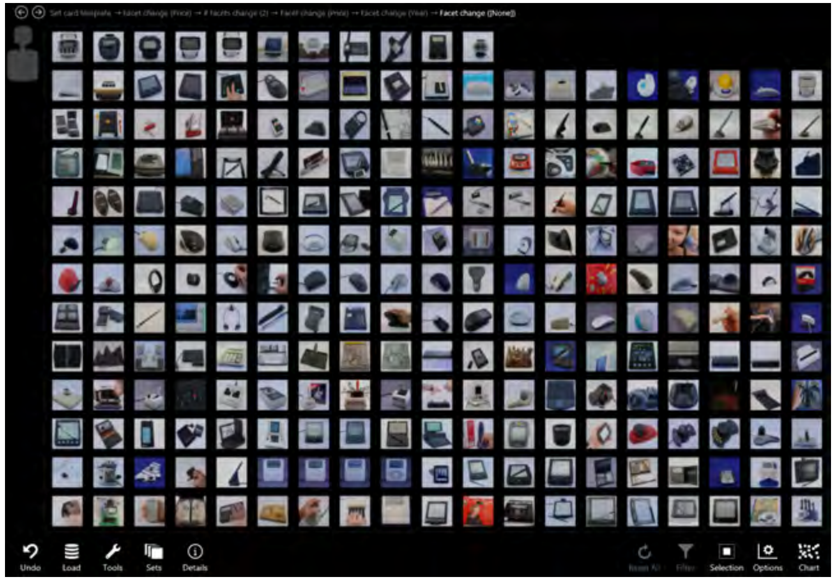
Layout

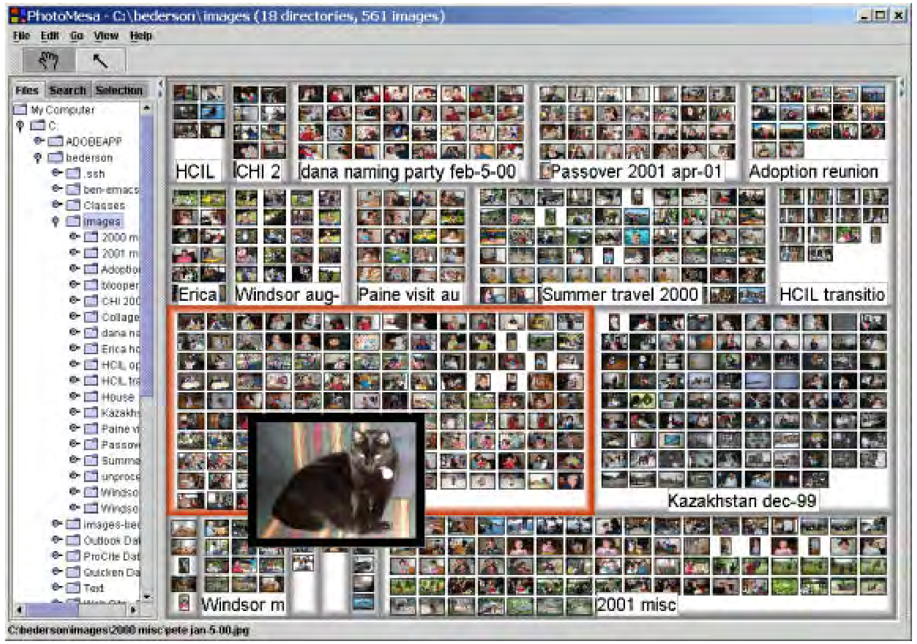


Layout
Overlapping
Space-filling
Map
jittering
Subdivision
Partition
Packing
Disjoint
1D
fillX, fillY
2D
Treemap
Grid
Physicalization

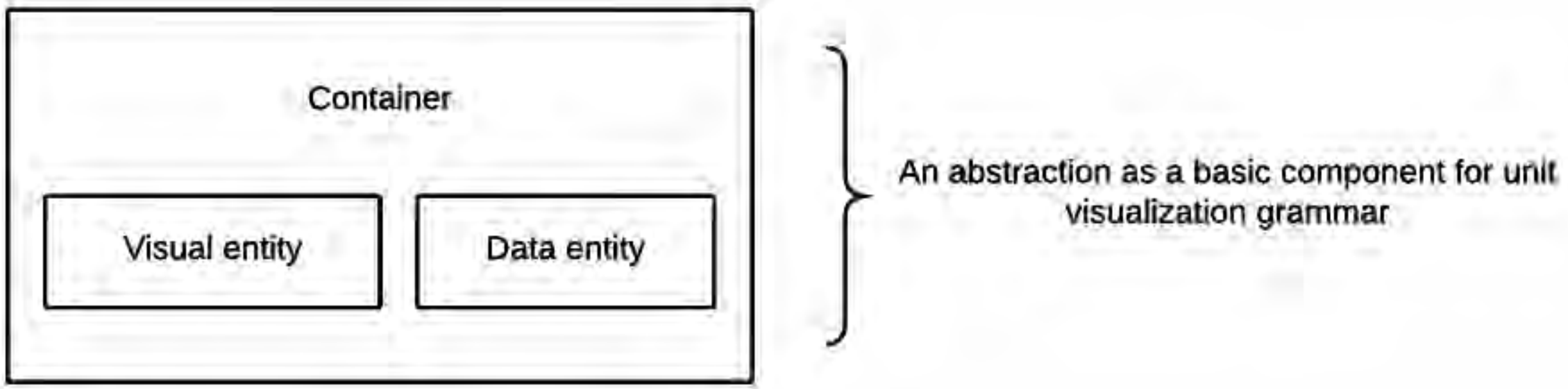
Unit representation



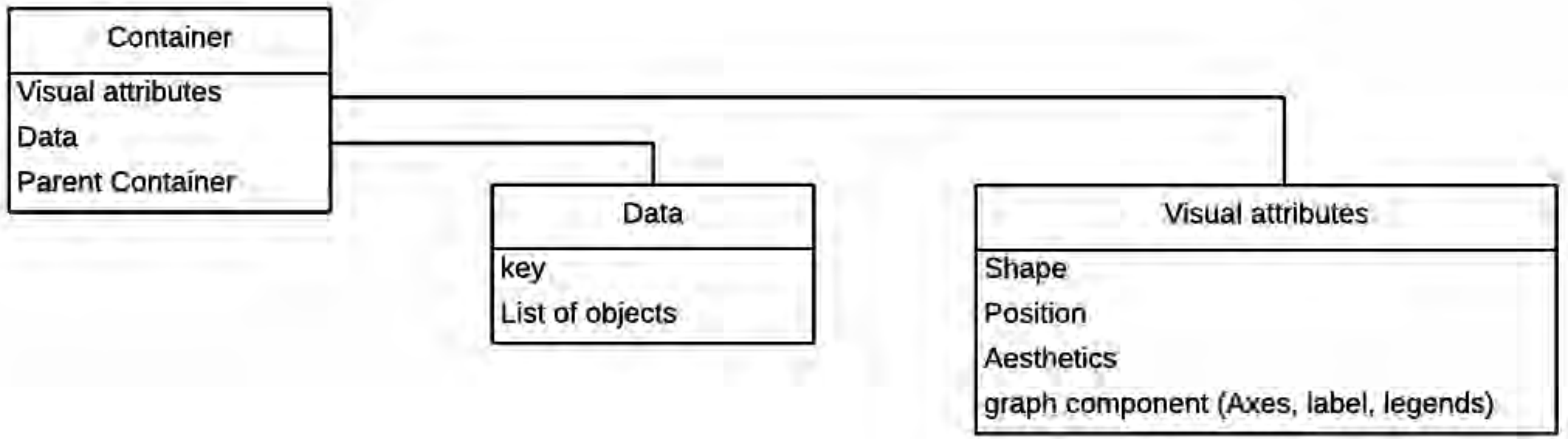




A Grammar for Unit Visualizations Process

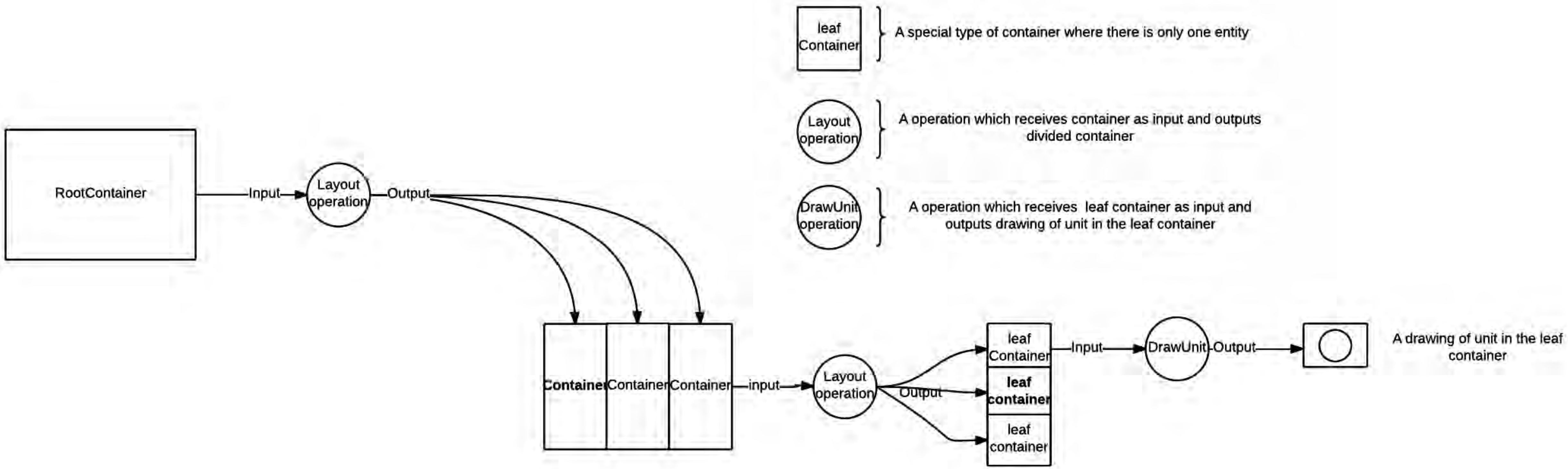


Container

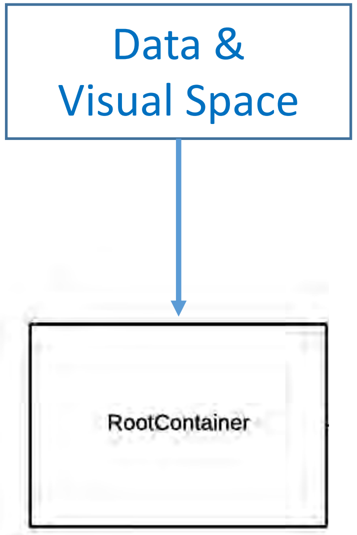


Container
Visual attributes
Data
Parent Container
Data
key
List of objects
Visual attributes
Shape
Position
Aesthetics
graph component (Axes, label, legends)

Process


leaf Container
A special type of container where there is only one entity
Layout operation
A operation which receives container as input and outputs divided container
DrawUnit operation
A operation which receives leaf container as input and outputs drawing of unit in the leaf container
RootContainer
Input
Layout operation
Output
Container
Container
Container
input
Layout operation
Output
leaf Container
leaf container
leaf container
Input
DrawUnit
Output
A drawing of unit in the leaf container

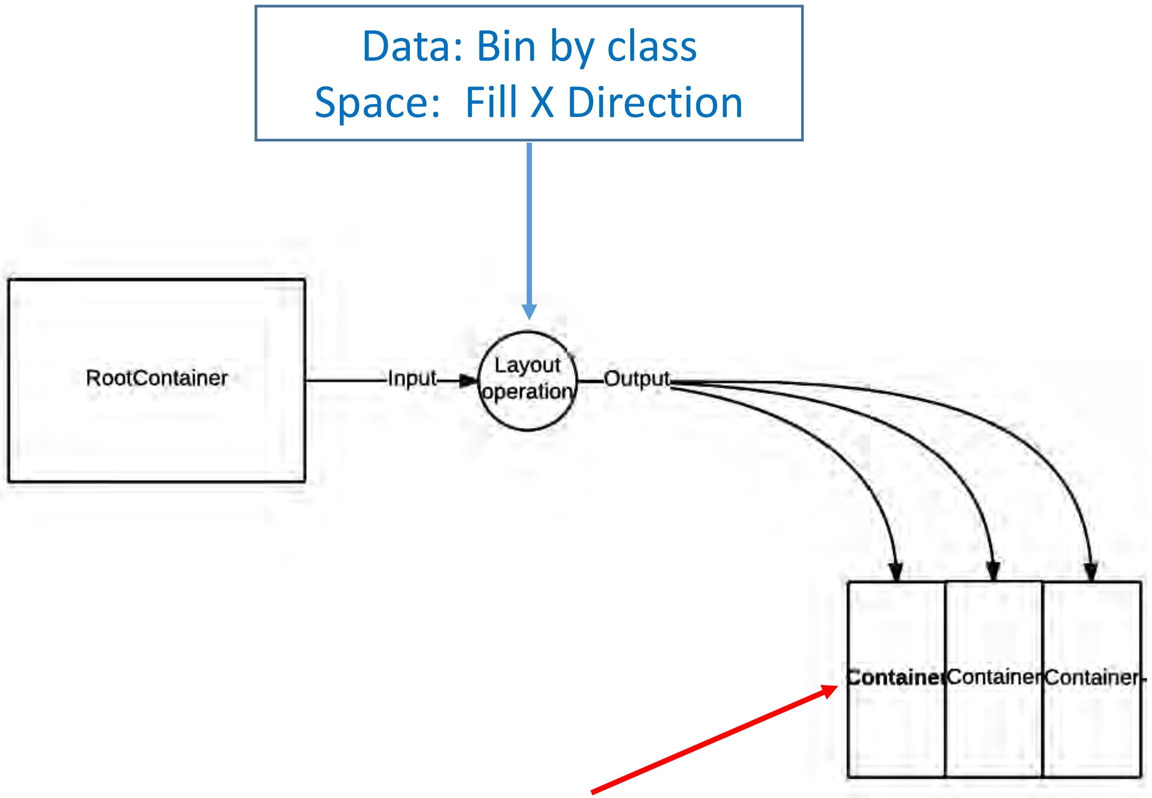
Process



All the passenger in the Titanic

Rect width 640px, height 480px

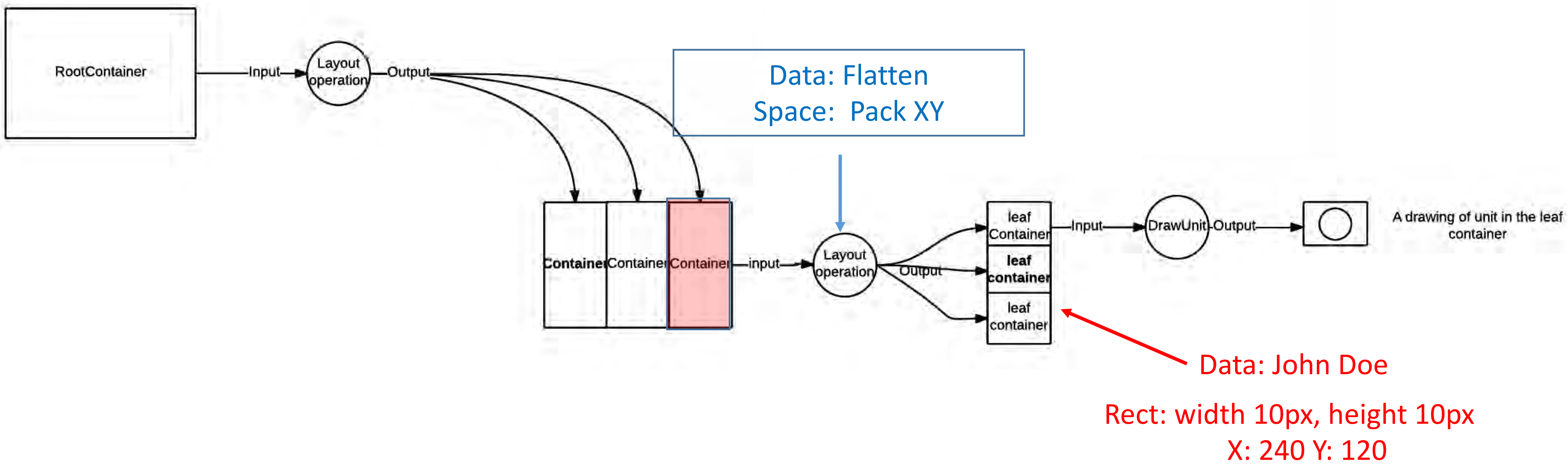
Process



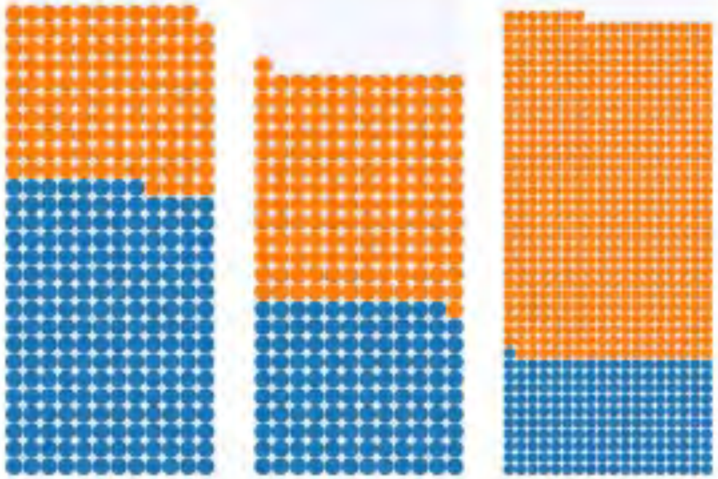
Data: 1st class passenger

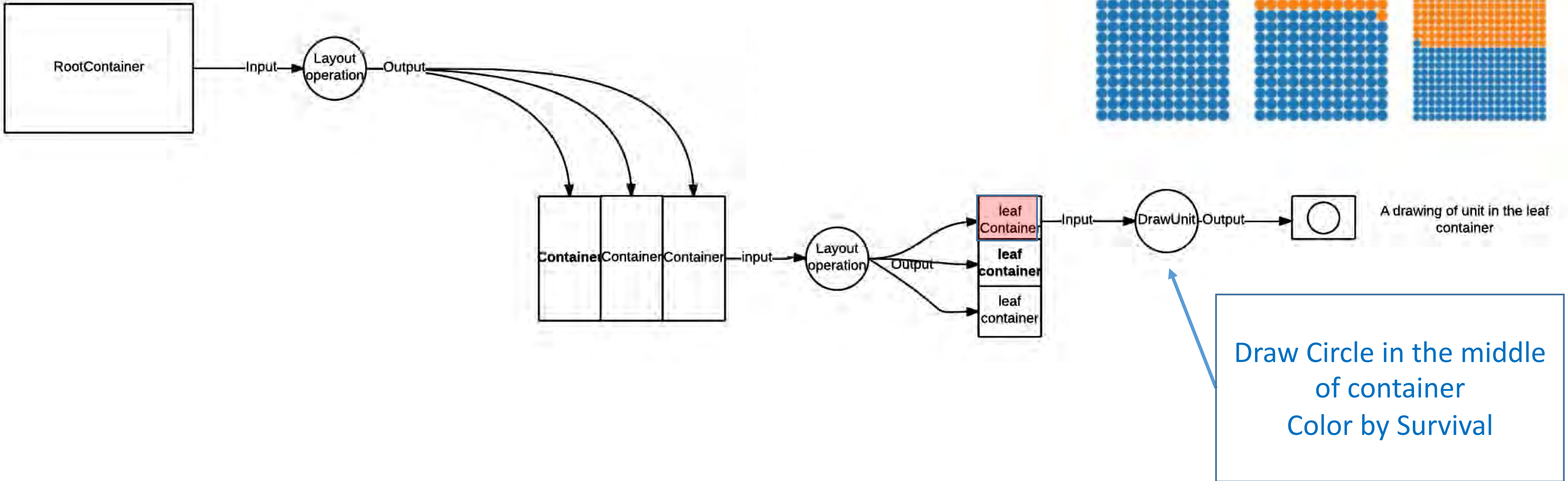
Rect: width 180px, height 480px
X: 0px Y: 0px

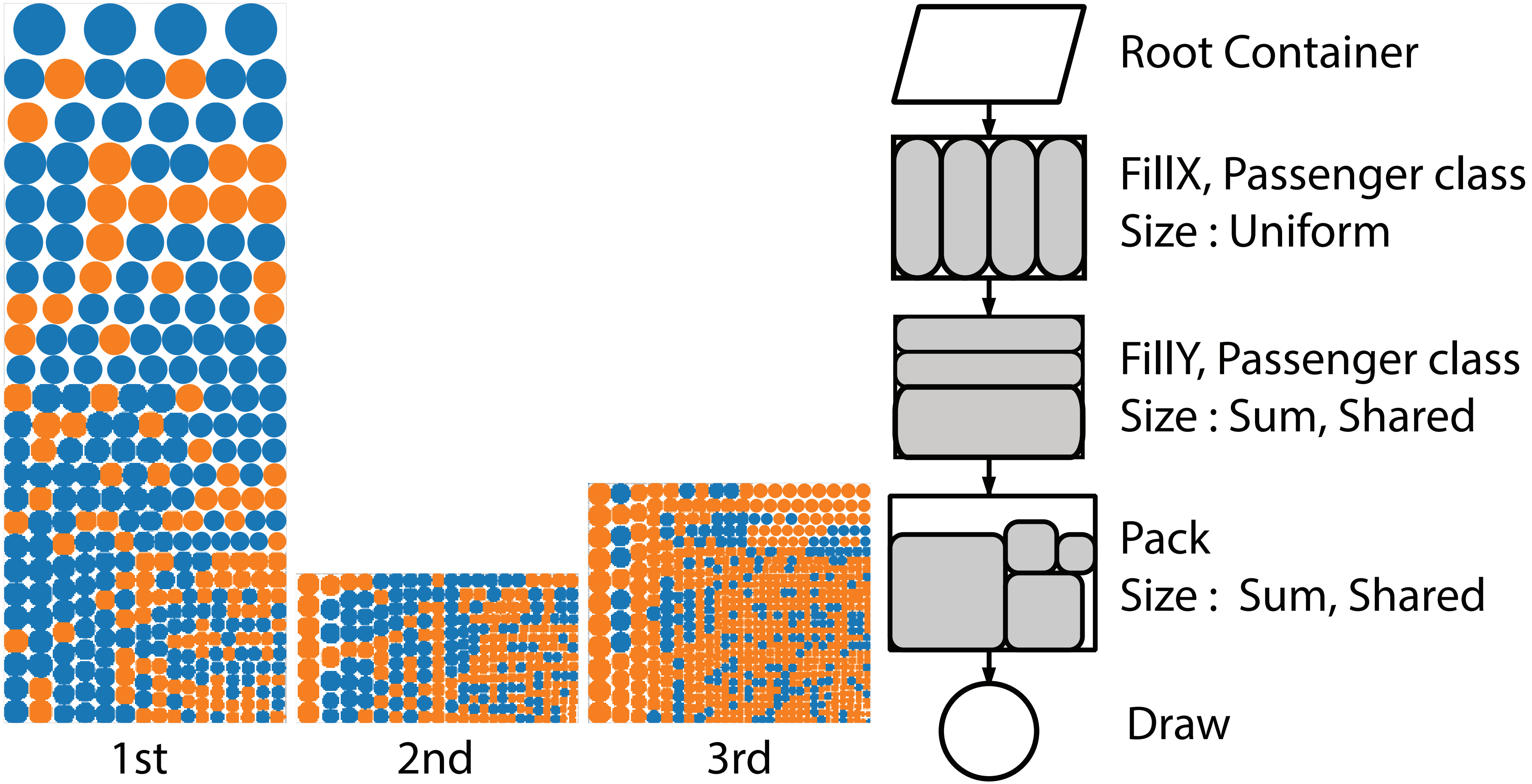
Process

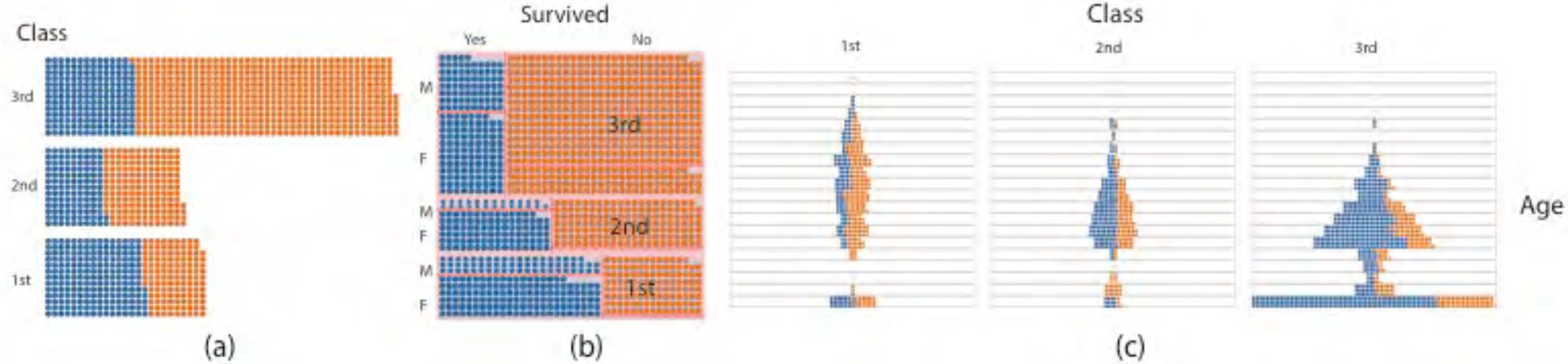

RootContainer
Input
Layout operation
Output
Data: Flatten
Space: Pack XY
Container
Container
Container
input
Layout operation
Output
leaf Container
leaf container
leaf container
Input
DrawUnit
Output
A drawing of unit in the leaf container
Data: John Doe
Rect: width 10px, height 10px
X: 240 Y: 120

Process




RootContainer
Input
Layout operation
Output
Container
Container
Container
input
Layout operation
Output
leaf Container
leaf container
leaf container
Input
DrawUnit
Output
A drawing of unit in the leaf container
Draw Circle in the middle of container
Color by Survival


1st
2nd
3rd
Root Container
FillX, Passenger class
Size : Uniform
FillY, Passenger class
Size : Sum, Shared
Pack
Size : Sum, Shared
Draw


Class
3rd
2nd
1st
(a)
Survived
Yes
No
M
F
M
F
M
F
3rd
2nd
1st
(b)
Class
1st
2nd
3rd
Age
(c)

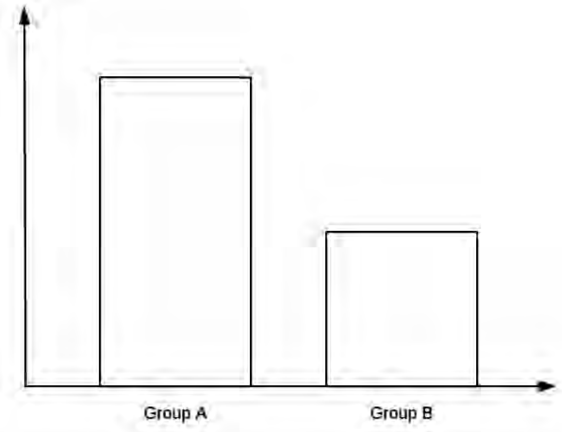
Conclusion

WHAT IS UNIT VISUALIZATION PROCESS?
WHY IT IS IMPORTANT?
HOW TO DRAW?

Aggregated Visualization

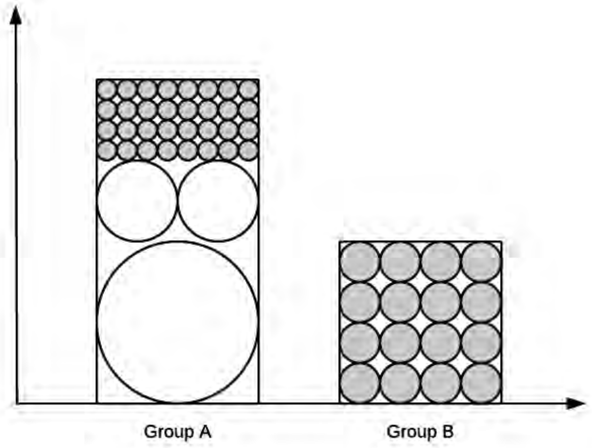
	Group	Income
Bill Gates	A	60
Steve Jobs	A	30
John Doe	B	2
...	...	...

	Total Income
Group A	60
Group B	30



Unit Visualization

	Group	Income
Bill Gates	A	60
Steve Jobs	A	30
John Doe	B	2
...	...	...

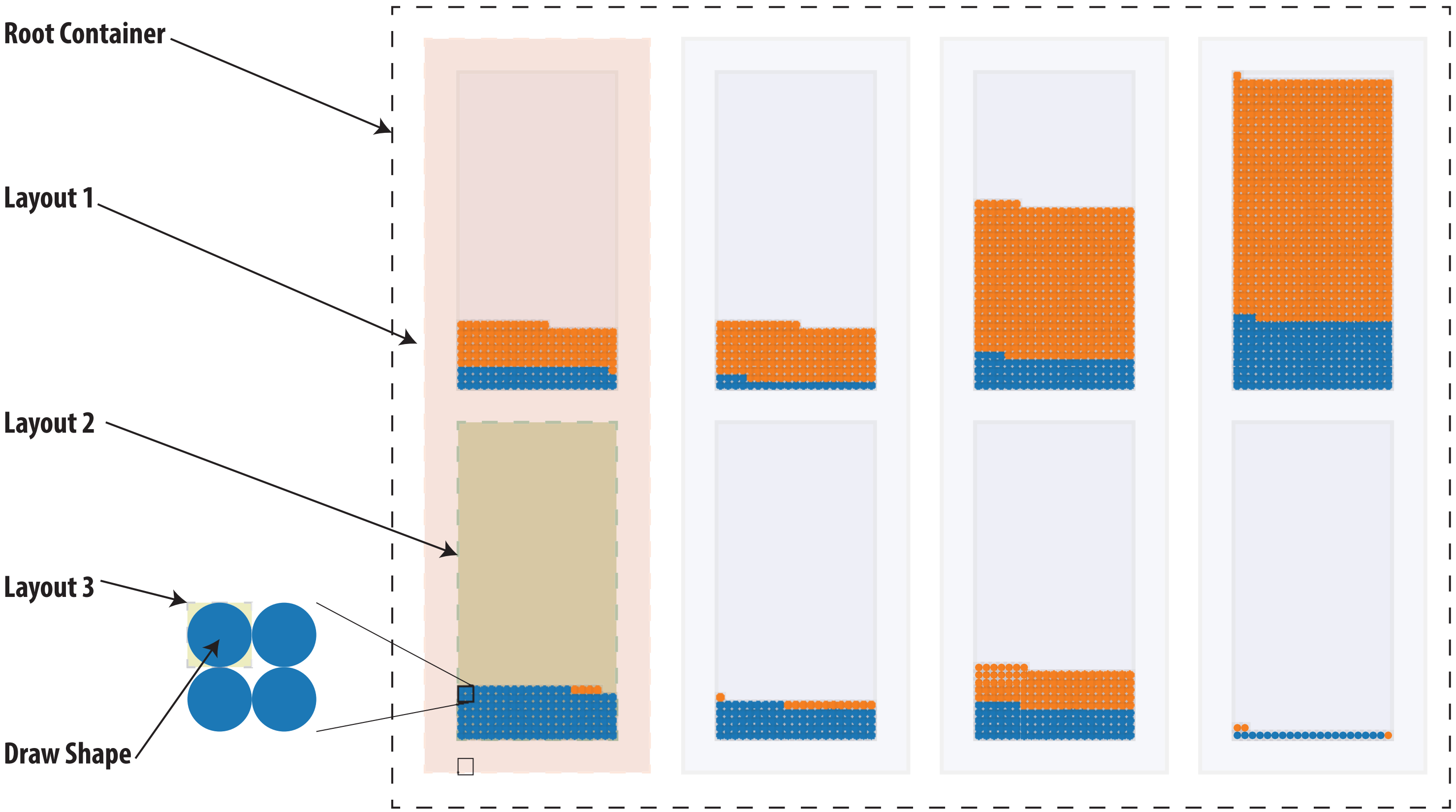


Benefit of Unit Visualizations Process

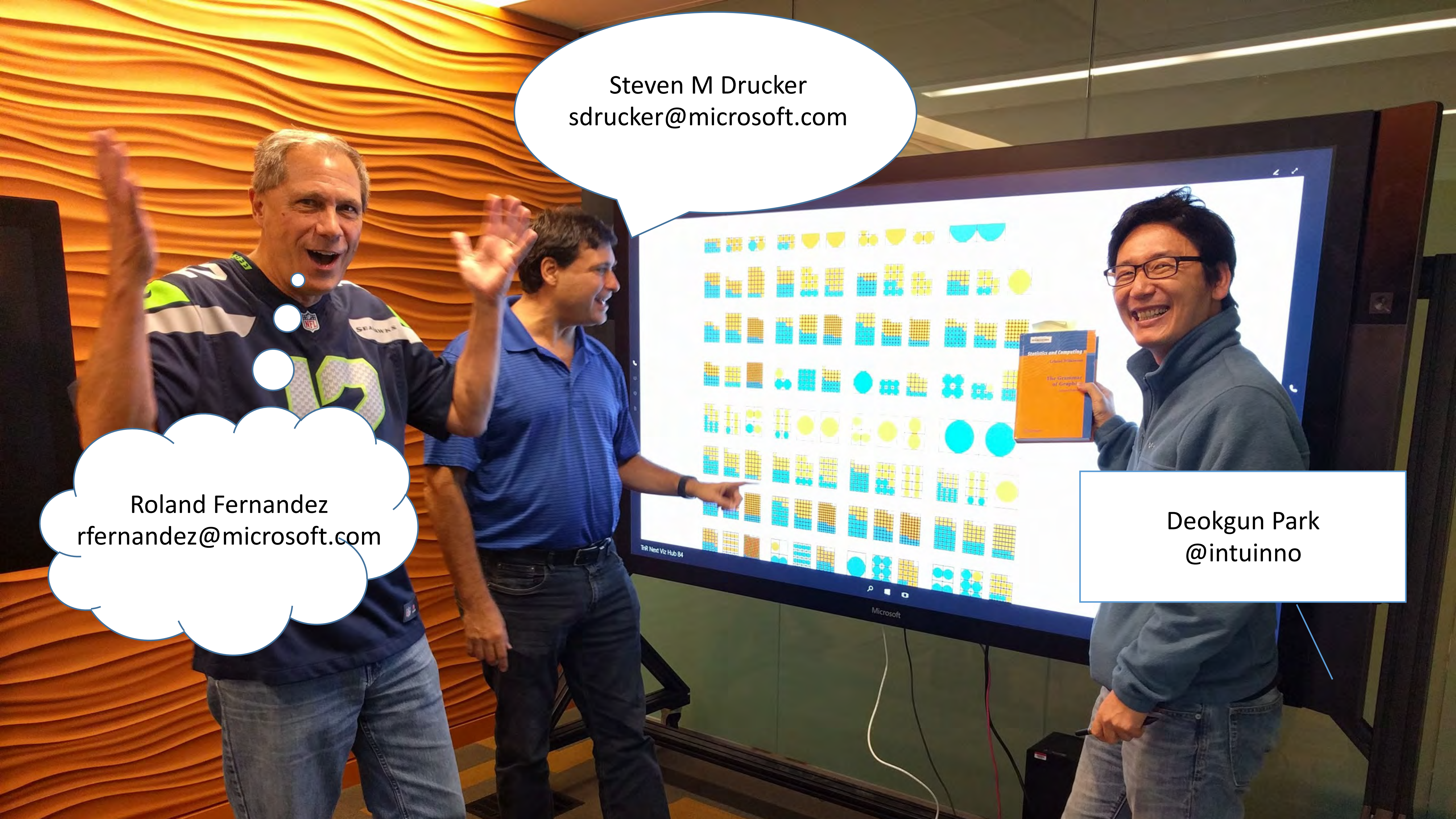
- Deliver More information
- Provides Natural format for Perception
- Enables Physical interactions

A Grammar for Unit Visualization Process

```
{
 "data": "data/titanic.csv",
 "width": 320, "height": 240,  "padding": {..},
 "layouts": [{
  "name": "layout1",
  "type": "gridxy",
  "subgroup": {  "type": "groupby",  "key": "Class" },
  "aspect_ratio": "fillX",
  "size": { "type": "uniform",  "isShared": false },
  "direction": "LRBT",  "align": "LB",
  "margin": { ... },  "padding": {... }
 }, {
  "type": "gridxy",
  "subgroup": { "type": "groupby", "key": "Sex", "isShared": true },
  "aspect_ratio": "fillY",
  "size": { "type": "uniform", "isShared": true },
 }, {
  "subgroup": { "type": "flatten" },
  "aspect_ratio": "square",
  "size": { "type": "uniform", "isShared": true },
  "sort": { "key": "Survived"}
 }],
 "mark": {
  "shape": "circle",
  "color": { "key": "Survived", "type": "categorical" },
  "size": { "type": "max",  "isShared": false },
  "isColorScaleShared": true
 }
}
```




TEAM WORK


Steven M Drucker
sdrucker@microsoft.com
Roland Fernandez
rfernandez@microsoft.com
Deokgun Park
@intuinno
Microsoft

Thank you.
Questions?

https://intuinno.github.io/unit/#/

Park, Deokgun, et al. "ATOM: A Grammar for Unit Visualizations." *IEEE Transactions on Visualization and Computer Graphics* (2017).

ATOM: A Grammar for Unit Visualizations

Deokgun Park, Steven M. Drucker, Roland Fernandez, and Niklas Elmqvist, *Senior Member, IEEE*

Abstract—Unit visualizations are a family of visualizations where every data item is represented by a unique visual mark—a visual *unit*—during visual encoding. For certain datasets and tasks, unit visualizations can provide more information, better match the user's mental model, and enable novel interactions compared to traditional aggregated visualizations. Current visualization grammars cannot fully describe the unit visualization family. In this paper, we characterize the design space of unit visualizations to derive a grammar that can express them. The resulting grammar is called ATOM, and is based on passing data through a series of layout operations that divide the output of previous operations recursively until the size and position of every data point can be determined. We evaluate the expressive power of the grammar by both using it to describe existing unit visualizations, as well as to suggest new unit visualizations.

Index Terms—Visualization grammar, unit visualizations, declarative specification.

1 Introduction

Visualization encodes symbolic data into visual structures [1], and arguably the most straightforward way to do this is to use a direct mapping where each data item becomes a unique visual mark. Such visualizations strictly maintain the *identity* of each visual mark and its relation to a corresponding data item. Drucker and Fernandez use the term *unit visualizations* to refer to this family of visualization techniques, and prominent examples of such techniques include unit charts, dotplots, and scatterplots [2]. In contrast, visualizations based on data aggregation—such as barcharts, piecharts, or histograms—merge multiple data items into inseparable graphic entities [3]. While such data abstraction improves the scalability of the visual representation, it surrenders the identity property of the visual marks, making it impossible to distinguish individual data points in the visualization. Maintaining the identity property, on the other hand, allows for many novel interactions not possible using an aggregating visualization, such as querying individual data points, tracking their movement during transitions, and filtering on an item level. While many useful visualizations that exhibit these properties exist, to date, this type of visualization has not yet been classified as a unique category, and their design space has not been systematically explored.

In this paper, we address this gap in the literature by presenting ATOM, a high-level grammar for unit visualizations based on a structured exploration of their design space. ATOM uses a sequence of recursive layout operations that organize the output of previous operations until the size and position of each data point can be

This yields a number of previously unknown visualizations that may be useful to explore further, and proves that our grammar also has significant *generative power*.

The remainder of this paper is structured as follows: We first define and discuss unit visualizations and their difference from visualizations that use aggregation. We then review the literature on current unit visualizations and visualization grammars. This leads to our design space of unit visualizations and a grammar for describing them. We validate our work with several examples of existing as well as novel unit visualizations. Finally, we discuss the Atom grammar in contrast to existing visual grammars and derive guidelines for how to best use them. We close the paper with our conclusion and our plans for future work.

2 Aggregated vs. Unit Visualizations

We define *unit visualizations* as visualizations that maintain the *identity* property of its visual marks, i.e., where each visual mark is a unique entity that is associated with a corresponding unique data item. The identity property means that for every data item in the data table, there is a corresponding visual mark in its visualization. While the unit visualization family has not yet been properly categorized in the visualization field, there nonetheless exist several examples of effective unit visualizations, such as unit charts, dotplots, and scatterplots.

Maintaining the identity property can lead to visual clutter for large datasets. To combat this, many visualization techniques are based on data abstraction, such as aggregation, segmentation,

Human Computer Interaction Laboratory

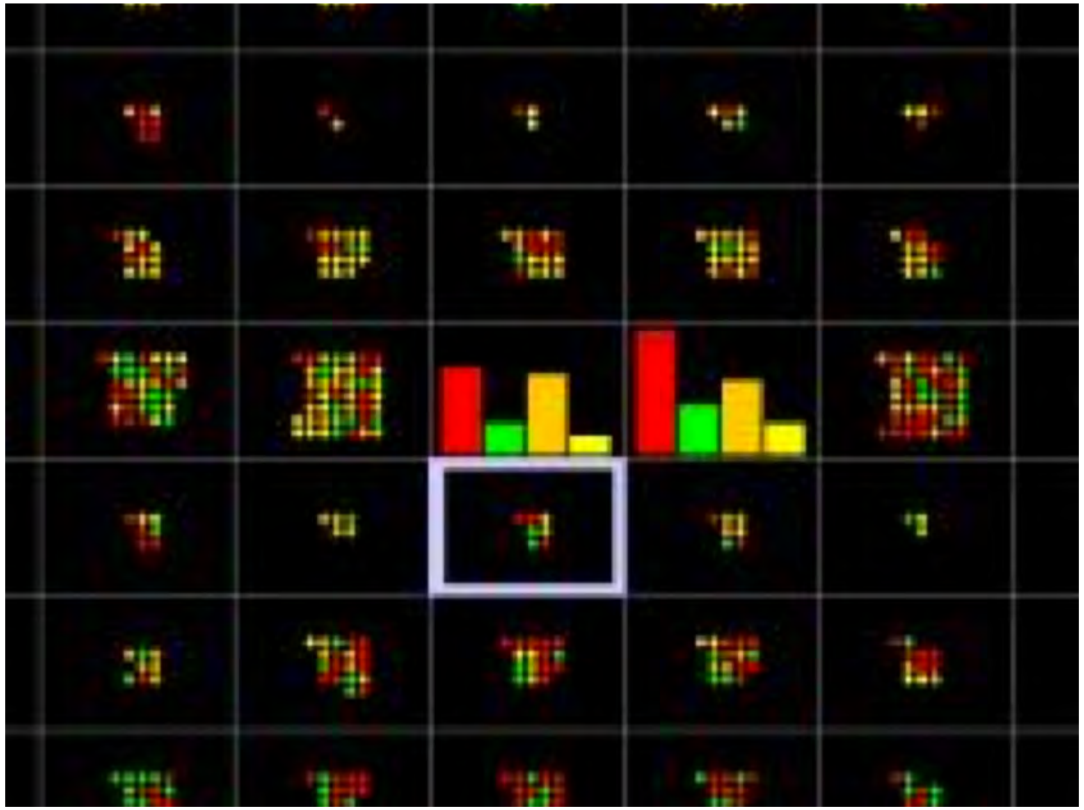
COMPUTER SCIENCE UNIVERSITY OF MARYLAND

UNIVERSITY OF MARYLAND

Microsoft Research

Limitations

- Scalability
  - Too many
  - Too little
- Visual Clutter



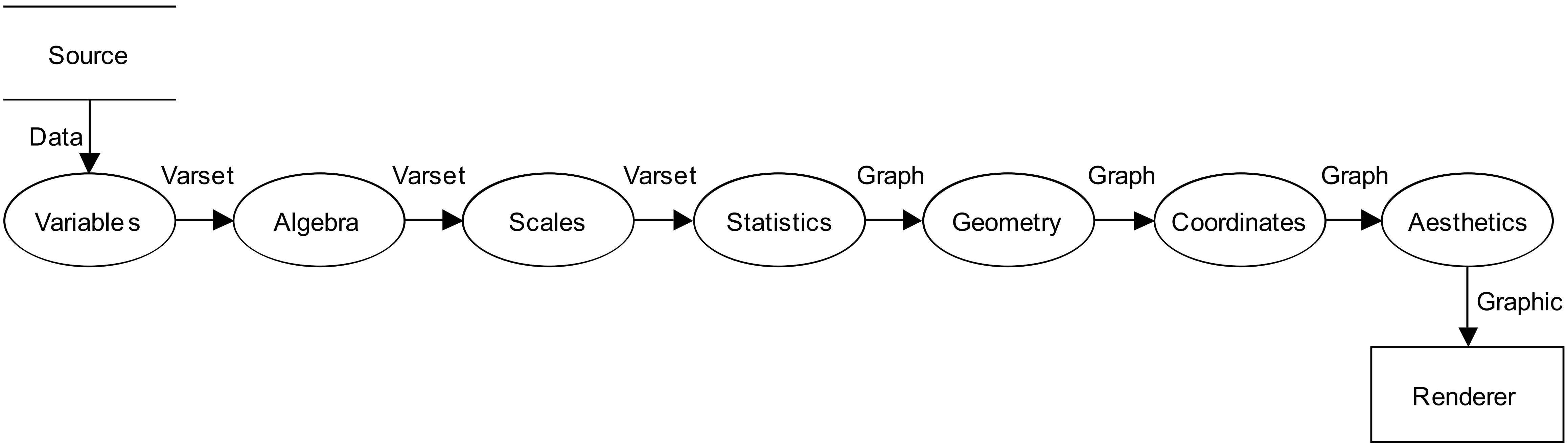
When to use UVP

- To show relative percentage or probability
- To check underlying distribution
- To check outliers
- For the democratic/casual visualizations
- At the beginning stage of exploratory Analysis

Diff with GG

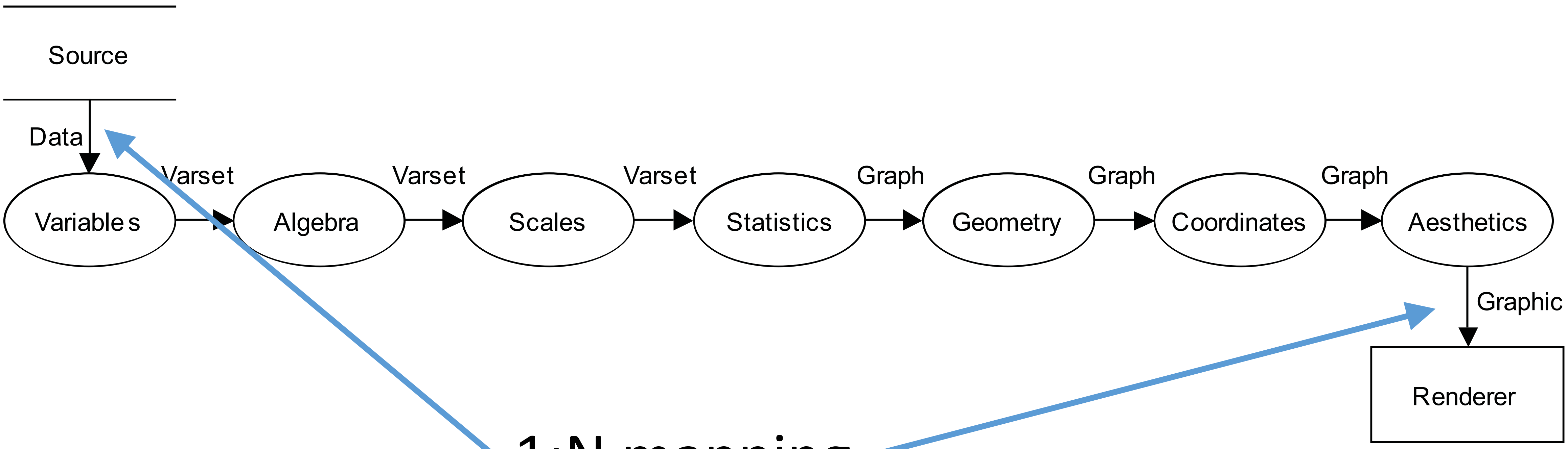
- Generalized Scale
  - Scale that returns Single numerical value is from Measurement theory
  - Scale for Visualizations returns subset of frame as an output
- Facet becomes obsolete
- Recursive iteration until leaf container level
- Collision modifier becomes first class citizen
  - Dodge/Stack
- Hierarchical sharing of visual properties

Visualization Process



From The Grammar of Graphics by Leland Wilkinson 2nd edition

Unit Visualization Process



1:N mapping

between source data and visual marks

Categorization based on mapping type

- N:1 -> Aggregated Visualization Process
- 1:1 -> Unit Visualization Process
- 1:N -> Unit Visualization Process
  - Tags
  - Isotypes
  - SPLOM