G53FIV: Fundamentals of Information Visualization

Lecture 13: Recap of Fundamentals

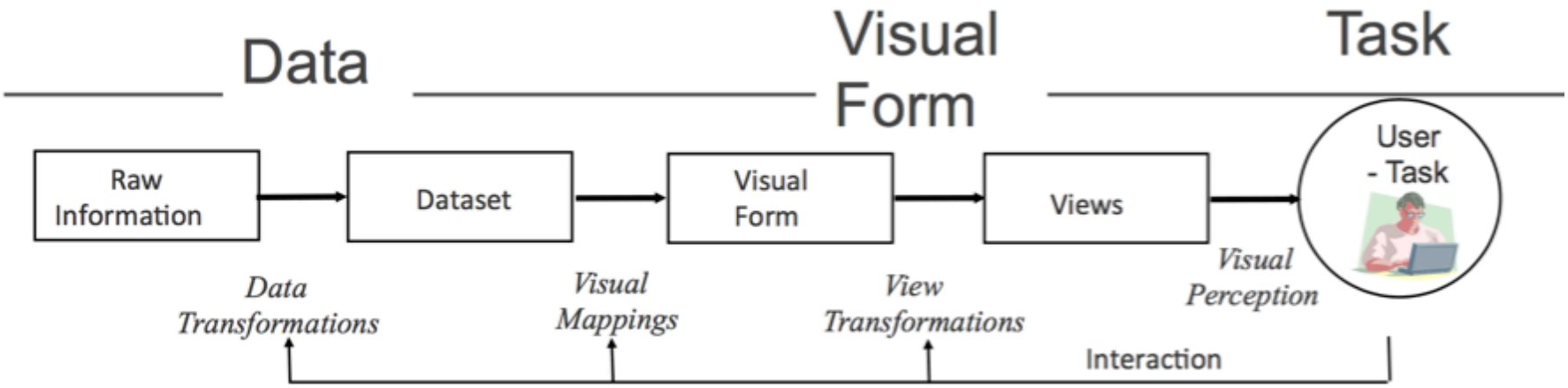
Ke Zhou
School of Computer Science
Ke.Zhou@nottingham.ac.uk

https://moodle.nottingham.ac.uk/course/view.php?id=68644

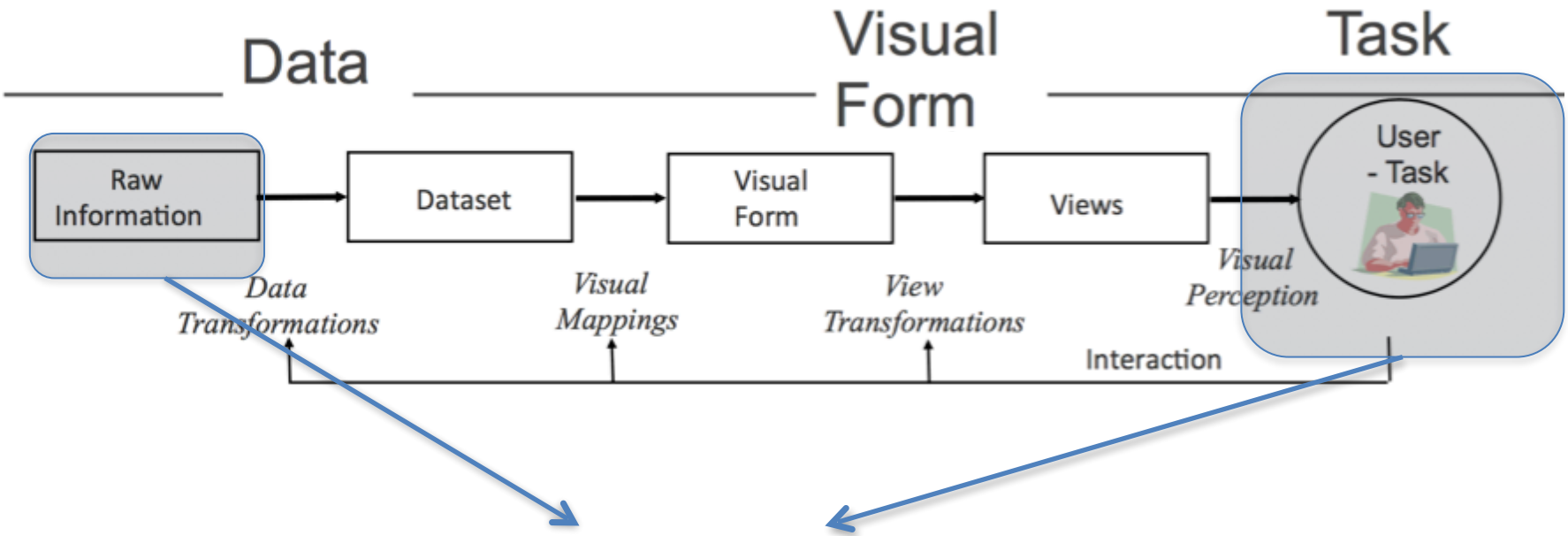
A Recap of Fundamentals

Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

Information Visualization



Information Visualization



Why IV? What are the values?

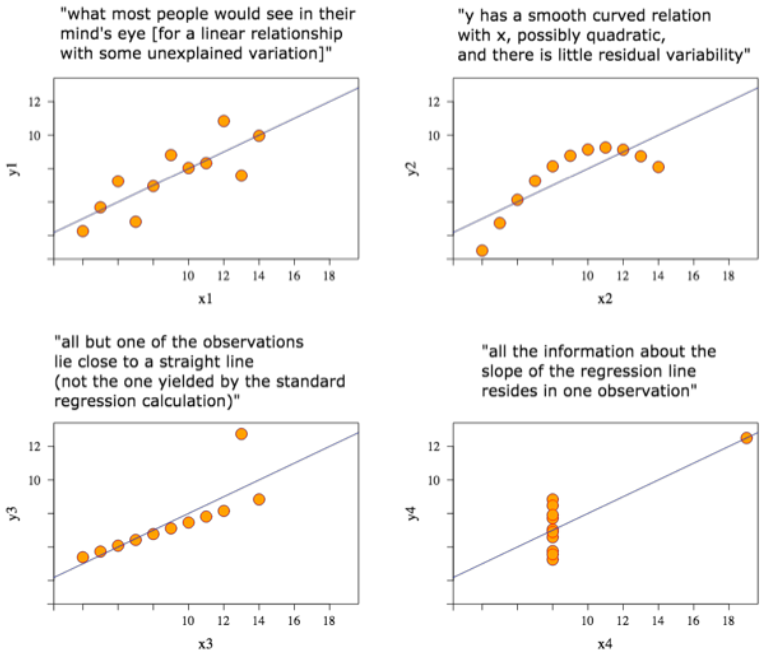
Information Overload

Objective

- Transform the data into information (understanding, insight) thus making it useful

Anscombe's Quartet

	Set A		Set B		Set C		Set D	
	X	Y	X	Y	X	Y	X	Y
0	10	8.04	10	9.14	10	7.46	8	6.58
1	8	6.95	8	8.14	8	6.77	8	5.76
2	13	7.58	13	8.74	13	12.74	8	7.71
3	9	8.81	9	8.77	9	7.11	8	8.84
4	11	8.33	11	9.26	11	7.81	8	8.47
5	14	9.96	14	8.10	14	8.84	8	7.04
6	6	7.24	6	6.13	6	6.08	8	5.25
7	4	4.26	4	3.10	4	5.39	19	12.50
8	12	10.84	12	9.13	12	8.15	8	5.56
9	7	4.82	7	7.26	7	6.42	8	7.91
10	5	5.68	5	4.74	5	5.73	8	6.89
mean	9.00	7.50	9.00	7.50	9.00	7.50	9.00	7.50
std	3.32	2.03	3.32	2.03	3.32	2.03	3.32	2.03
corr	0.82		0.82		0.82		0.82	
lin. reg.	$y = 3.00 + 0.500x$		$y = 3.00 + 0.500x$		$y = 3.00 + 0.500x$		$y = 3.00 + 0.500x$	



[Anscombe, 1973]

The Best of Both Sides

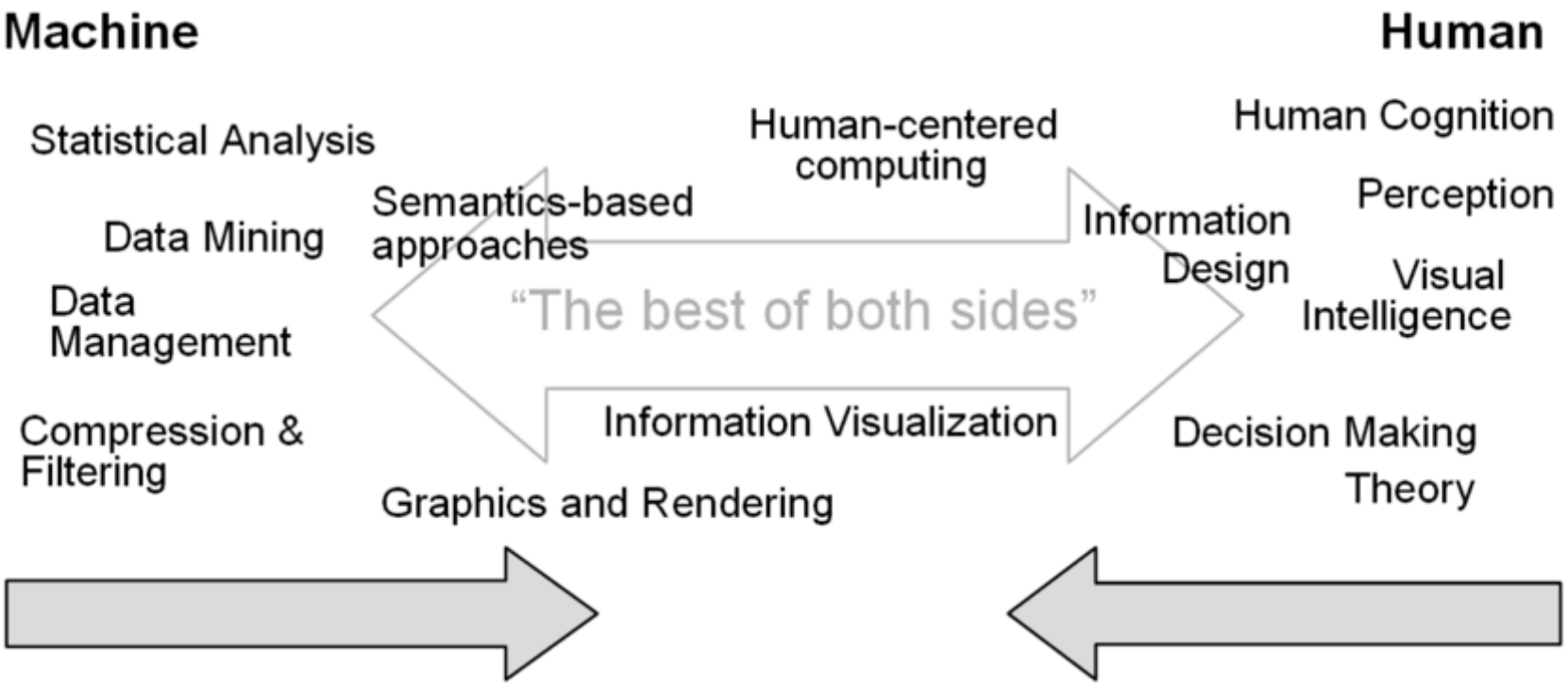


Fig. 2. Visual analytics integrates scientific disciplines to improve the division of labor between human and machine.

Visual Analytics: Definition, Process, and Challenges. Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, Guy Melançon, 2008.

Key Values of Visualizations

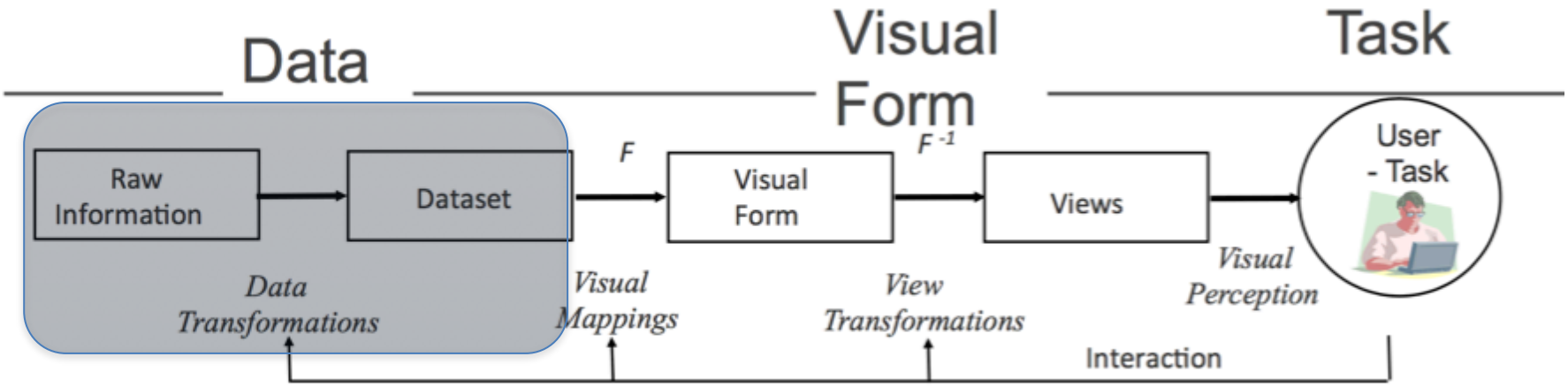
- **Record** information
  - Blueprints, photographs, seismographs, ...
- **Communicate** information to others
  - Share and persuade
  - Collaborate and revise
- Analyze data to **support reasoning**
  - Find patterns / Discover errors in data
  - Expand memory
  - Develop and assess hypotheses

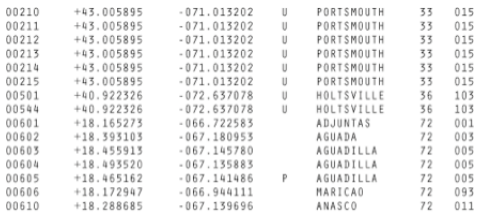
Data and Initial Tasks: Before the Actual Visualization

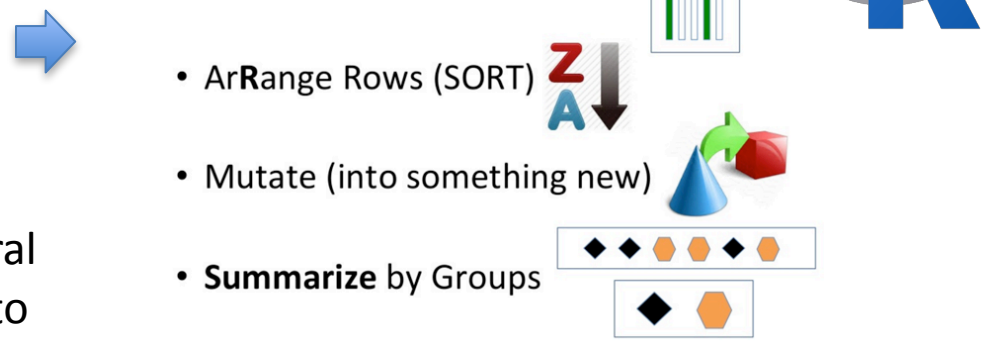
- Pick a dataset of your interest
- Pose the initial questions/tasks that you would like to answer/ accomplish
- Assess the fitness of the data
- **Visualization**
- Refine your questions/tasks

Remember our course work and the house price case study?

Information Visualization



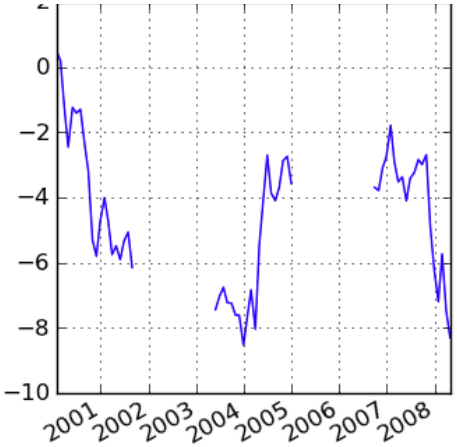




- **Data transformation**: create a structural model (schema), mapping raw data into data tables

Data Processing

- Data cleaning and filtering
  - for quality control
  - Remove (Outlier, missing data)
  - Modify (conversion of format, etc.)



- Data adjustment
  - Depends on your task and questions to ask
  - Relational algebra:
    - e.g. Aggregation, mean, sort, projection
  - Reformatting and Integration

Reading: https://en.wikipedia.org/wiki/Relational_algebra

R is a tool for...

Data Manipulation

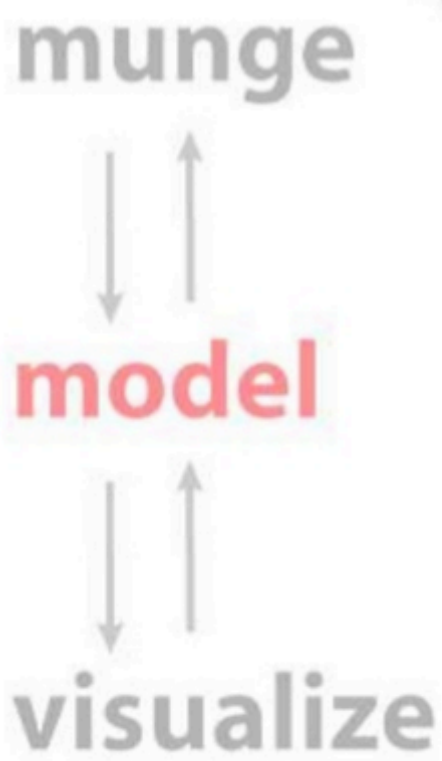
- connecting to data sources
- slicing & dicing data

Modeling & Computation

- statistical modeling
- numerical simulation

Data Visualization

- visualizing fit of models
- composing statistical graphics



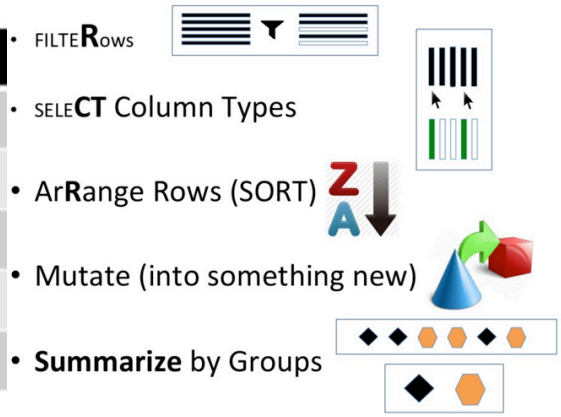
R Data Structures

	Linear	Rectangular
Homogeneous	vectors	matrices
Heterogeneous	lists	data frames*

dplyr

- dplyr takes the %>% operator and uses it to great effect for manipulating data frames
  - Works only with data frames
  - 5 basic "verbs" work for 90% of data manipulations manipulations

Verbs	What does it do?
`filter()`	Select a subset of ROWS by conditions
`arrange()`	Reorders ROWS in a data frame
`select()`	Select the COLUMNS of interest
`mutate()`	Create new columns based on existing columns (mutations!)
`summarise()`	Aggregate values for each group, reduces to single value

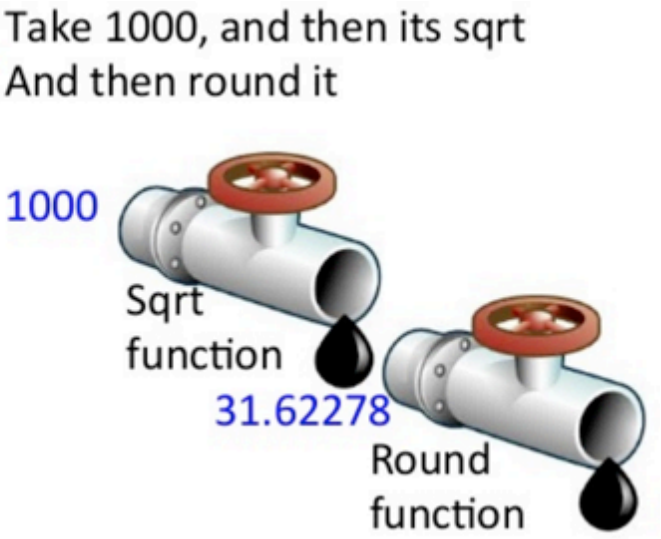


Pipe Operator

- Library(maggritr)
  - A R package launched on Jan 2014
  - A "magic" operator called the PIPE was introduced
  - %>%
  - i.e. "AND THEN", "PIPE TO"

```
round(sqrt(1000), 3)

library(magrittr)
1000 %>% sqrt %>% round()
1000 %>% sqrt %>% round(.,3)
```



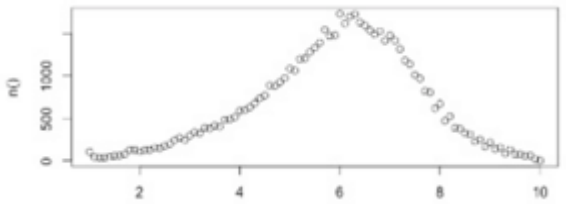
Chain the "Verbs" Together

- Chain them together

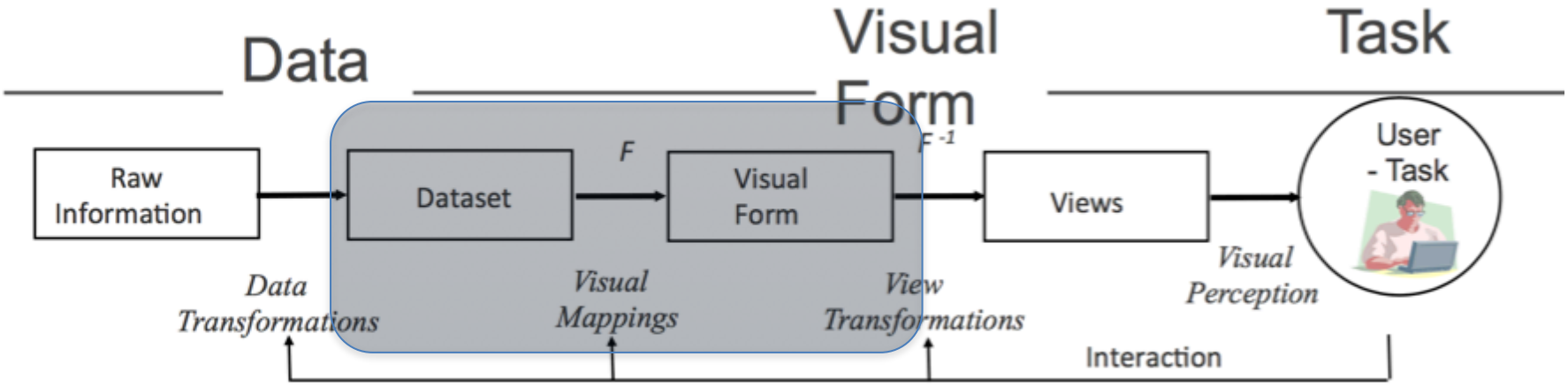
```
producers_nightmare <-
  filter(movies_df, !is.na(budget)) %>%
  mutate(costPerMinute = budget/length) %>%
  arrange(desc(costPerMinute)) %>%
  select(title, costPerMinute)
```

- Can also be fed to a "plot" command

```
movies %>%
  group_by(rating) %>%
  summarize(n()) %>%
  plot() # plots the histogram of movies by Each value of rating
```



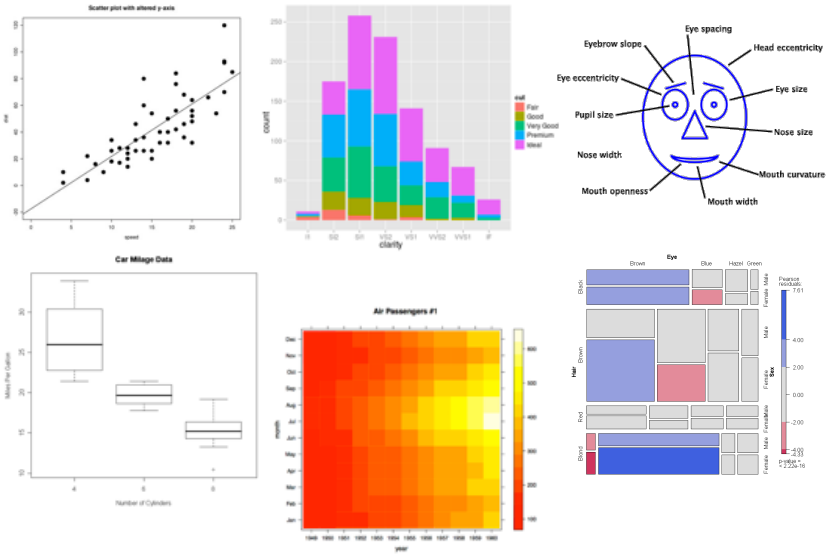
Information Visualization




```
00210    +43.005895    -071.013202    U    PORTSMOUTH    33    015
00211    +43.005895    -071.013202    U    PORTSMOUTH    33    015
00212    +43.005895    -071.013202    U    PORTSMOUTH    33    015
00213    +43.005895    -071.013202    U    PORTSMOUTH    33    015
00214    +43.005895    -071.013202    U    PORTSMOUTH    33    015
00215    +43.005895    -071.013202    U    PORTSMOUTH    33    015
```

	Nominal	Ordinal	Quantitative
Position	✓	✓	✓
Size	✓	✓	~
(Grey)Value	✓	✓	~
Texture	✓	~	✕
Color	✓	✕	✕
Orientation	✓	✕	✕
Shape	✓	✕	✕

✓ = Good
~ = OK
✕ = Bad



Visual mapping: create a visual spatial model, transforming data tables into visual structures

Data, Image and Design

Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

Data Models

	A	B	C	D	E
1	year	age	marst	sex	people
2	1850	0	0	1	1483789
3	1850	0	0	2	1450376
4	1850	5	0	1	1411067
5	1850	5	0	2	1359668
6	1850	10	0	1	1260099
7	1850	10	0	2	1216114
8	1850	15	0	1	1077133
9	1850	15	0	2	1110619
10	1850	20	0	1	1017281
11	1850	20	0	2	1003841
12	1850	25	0	1	862547
13	1850	25	0	2	799482
14	1850	30	0	1	730638
15	1850	30	0	2	639636
16	1850	35	0	1	588487
17	1850	35	0	2	505012
18	1850	40	0	1	475911
19	1850	40	0	2	428185
20	1850	45	0	1	384211
21	1850	45	0	2	341254
22	1850	50	0	1	321343
23	1850	50	0	2	286580
24	1850	55	0	1	194080

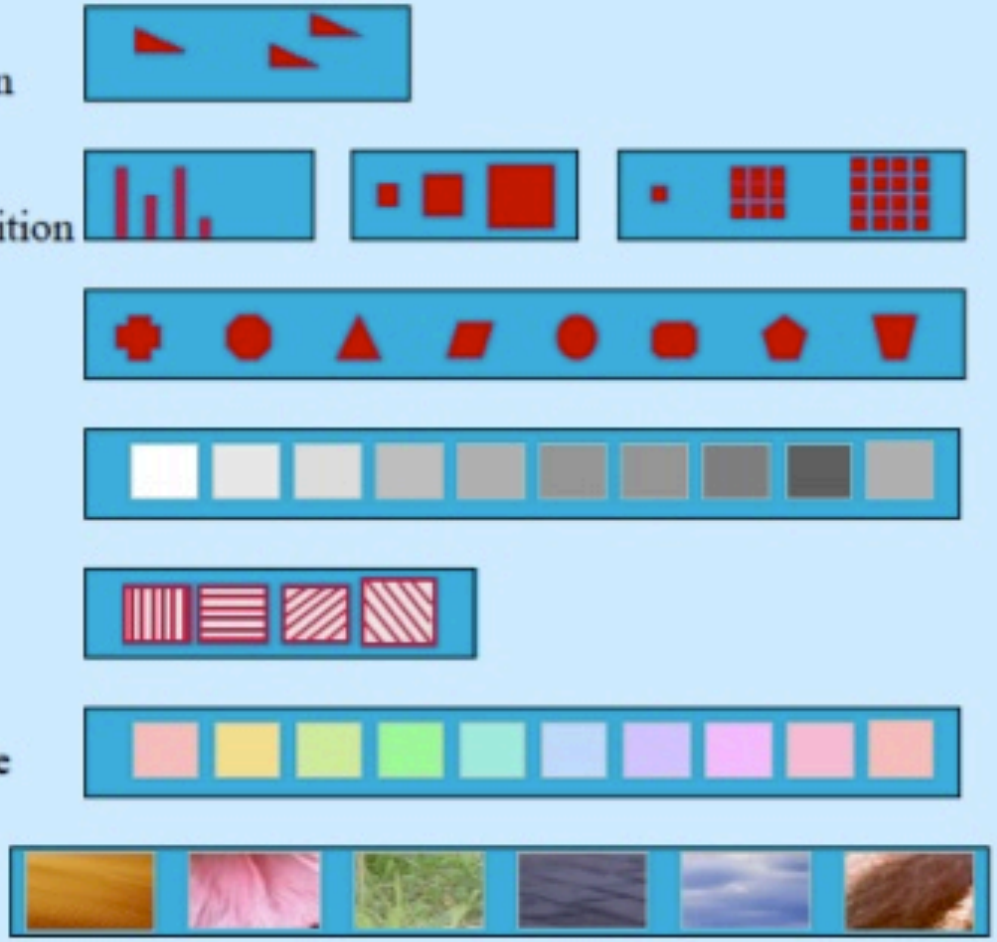
- Nominal, Ordinal, Quantitative?
- Dimension or Measure?

– Year	Q-Internal (O)	Dimension
– Age	Q-Ratio (O)	Depends
– Marital	N	Dimension
– Sex	N	Dimension
– People	Q-Ratio	Measure

Image: Visual Encoding Variables

Bertin's Semiology of Graphics (1967)

- **position**
  - changes in the x, y, (z) location
- **size**
  - change in length, area or repetition
- **shape**
  - infinite number of shapes
- **value**
  - changes from light to dark
- **orientation**
  - changes in alignment
- **colour**
  - changes in hue at a given value
- **texture**
  - variation in pattern
- **motion**



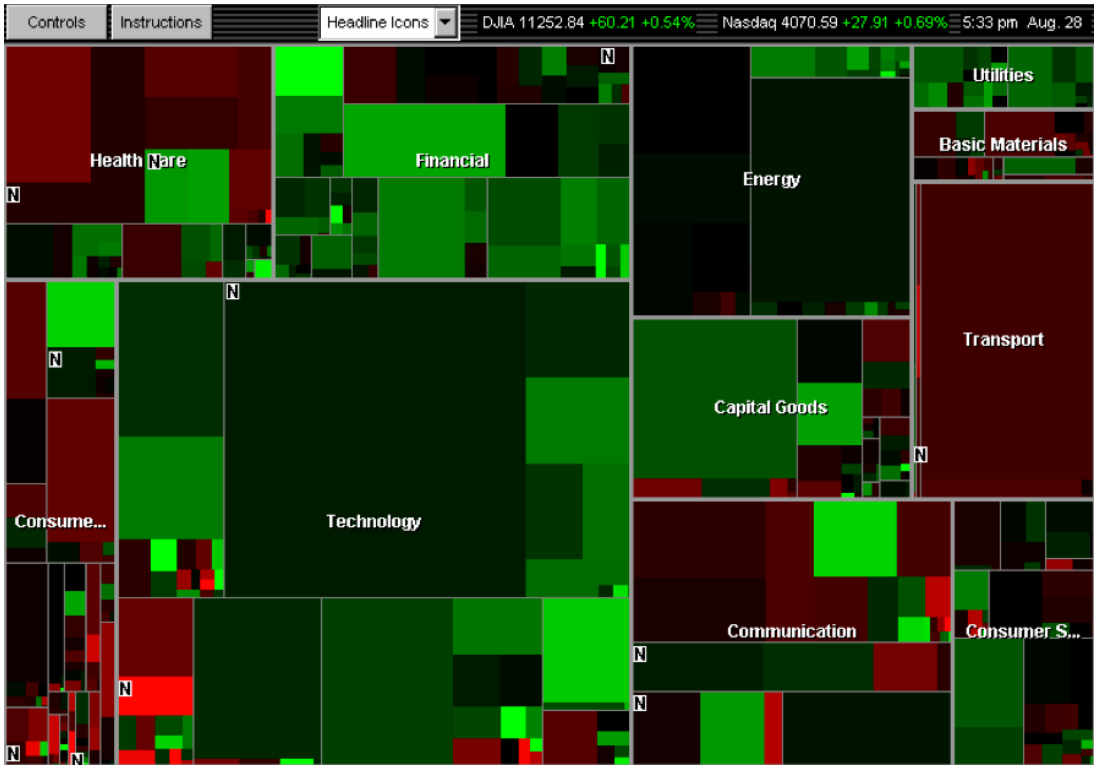
Graphic by: Sheelagh Carpendale

Levels of Organization

	Nominal	Ordinal	Quantitative
Position	✓	✓	✓
Size	✓	✓	~
(Grey)Value	✓	✓	~
Texture	✓	~	✗
Color	✓	✗	✗
Orientation	✓	✗	✗
Shape	✓	✗	✗

✓ = Good
~ = OK
✗ = Bad

Example: Map of the Market



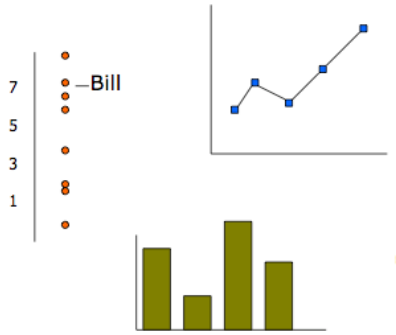
- Rectangle area: market cap (Q);
- Rectangle position: market sector (N)
- Color Hue: loss vs. gain (N, O)
- Color Value: magnitude of loss or gain (Q)

Graphs and Charts

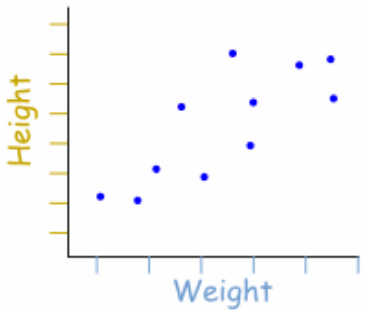
Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

Graphs

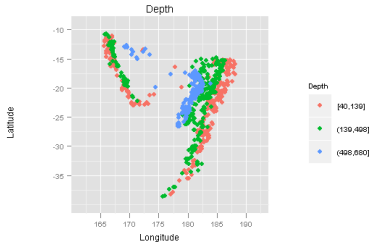
- Data Dimensions
  - 1 - Univariate data
  - 2 - Bivariate data
  - 3 - Trivariate data
  - >3 - Hypervariate data
- Data Types
  - Nominal, Ordinal, Quantitative
- Visualization Representations
  - Points, Lines, Bars, Boxes



Univariate



Bivariate

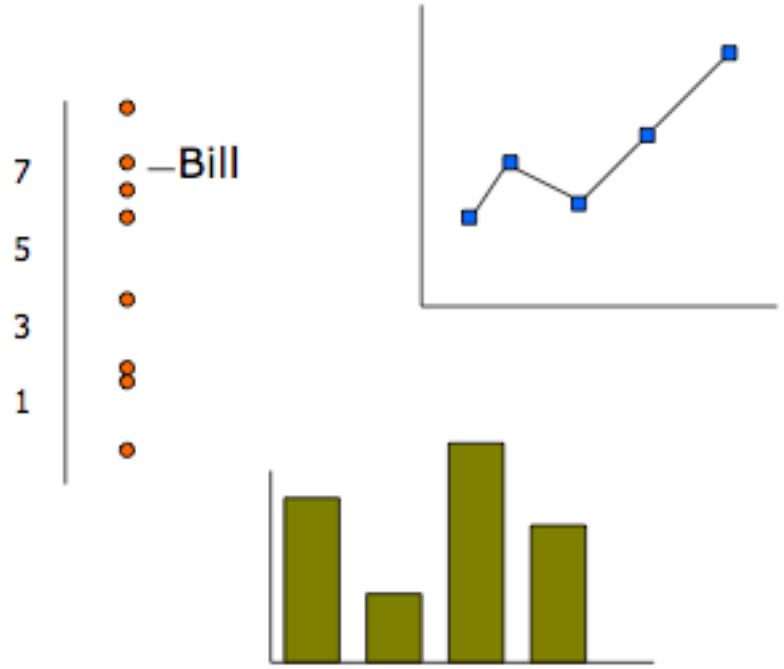


Trivariate

Univariate Data

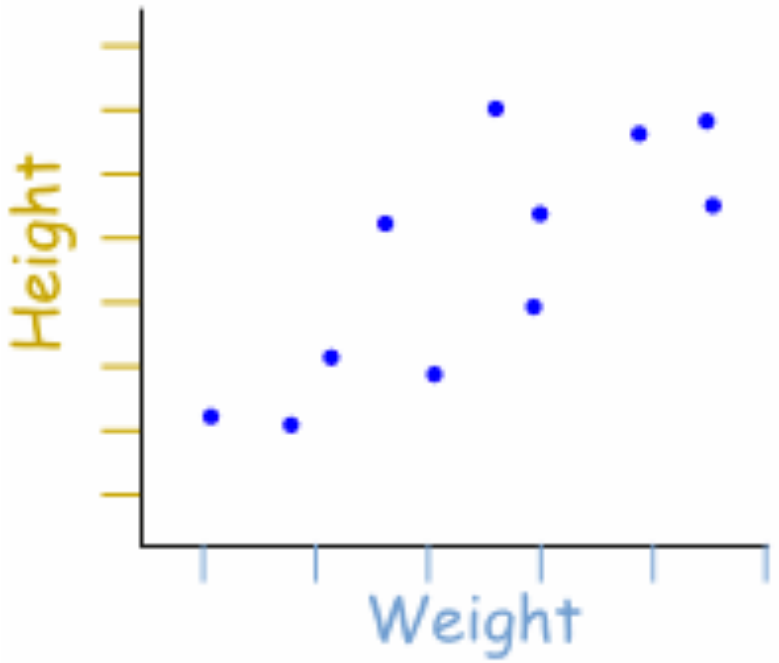
- In univariate representations, we often think of the data case as being shown along one dimension, and the value in another.

- Statistical view
  - Independent variable on x-axis (data case)
  - Track dependent variable along y-axis (value)



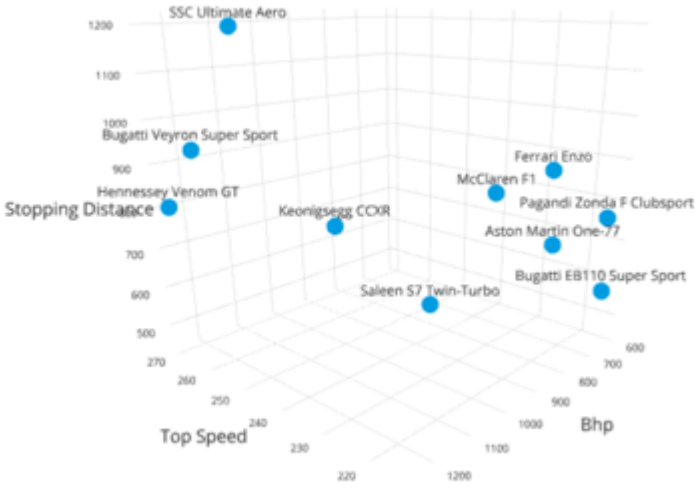
Bivariate Data

- Scatter plot is commonly used
- Each mark is now a data case
- Objective:
  - Two variables, want to see relationship
  - Is there a linear, curved or random pattern?

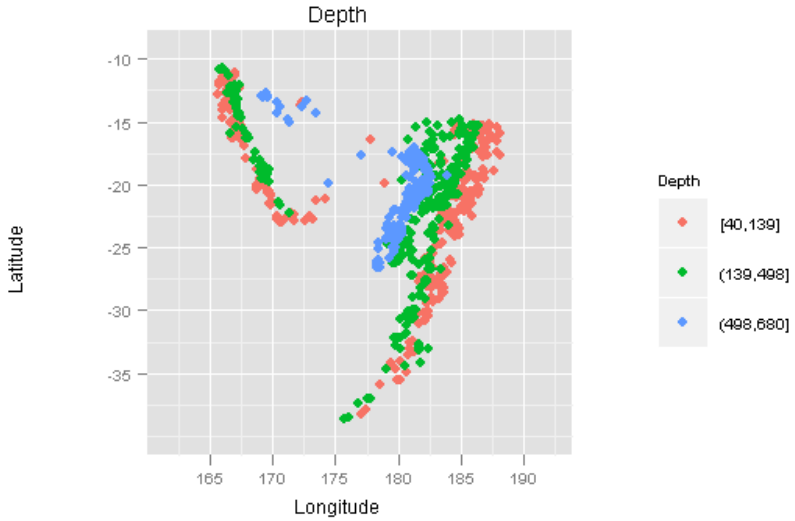


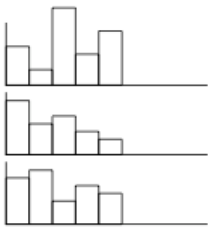
Trivariate Data

3D scatter plot



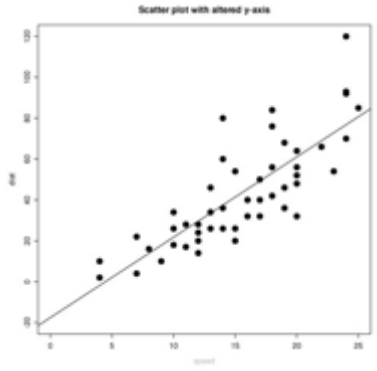
2D + mark property

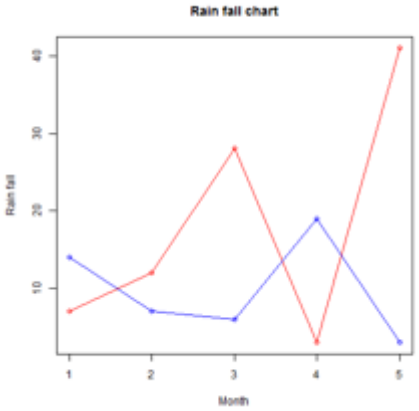


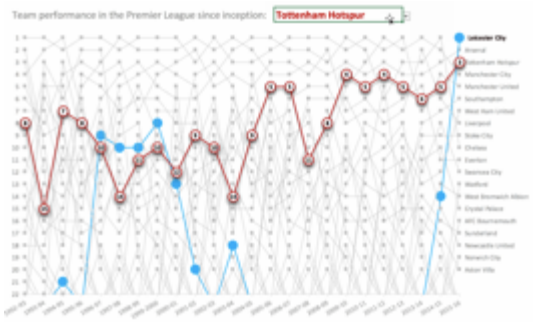


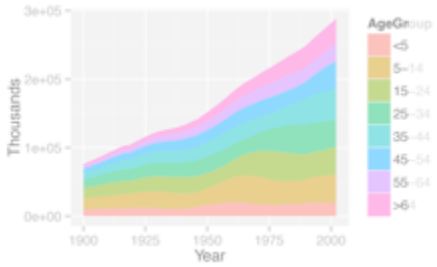
Represent each variable in its own explicit way

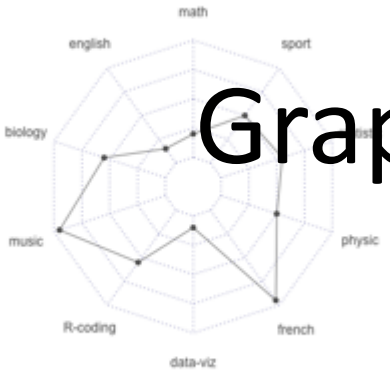
Graphs

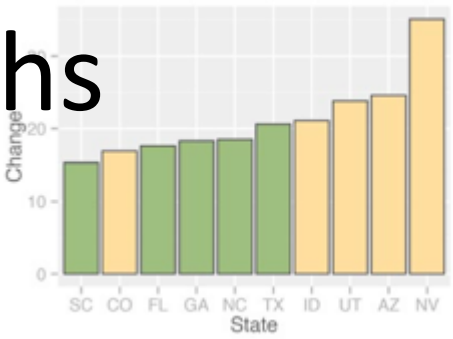


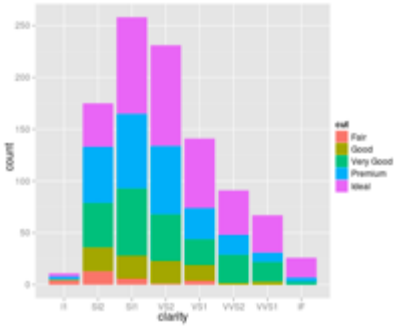


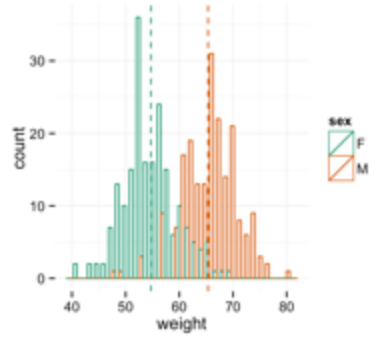


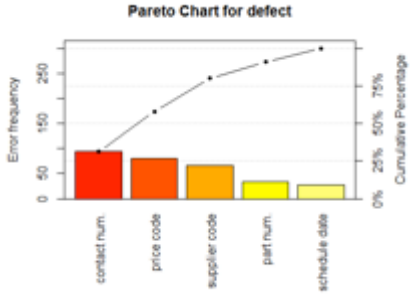


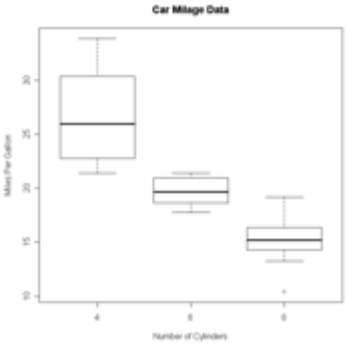


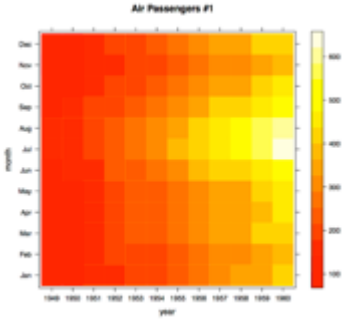












Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

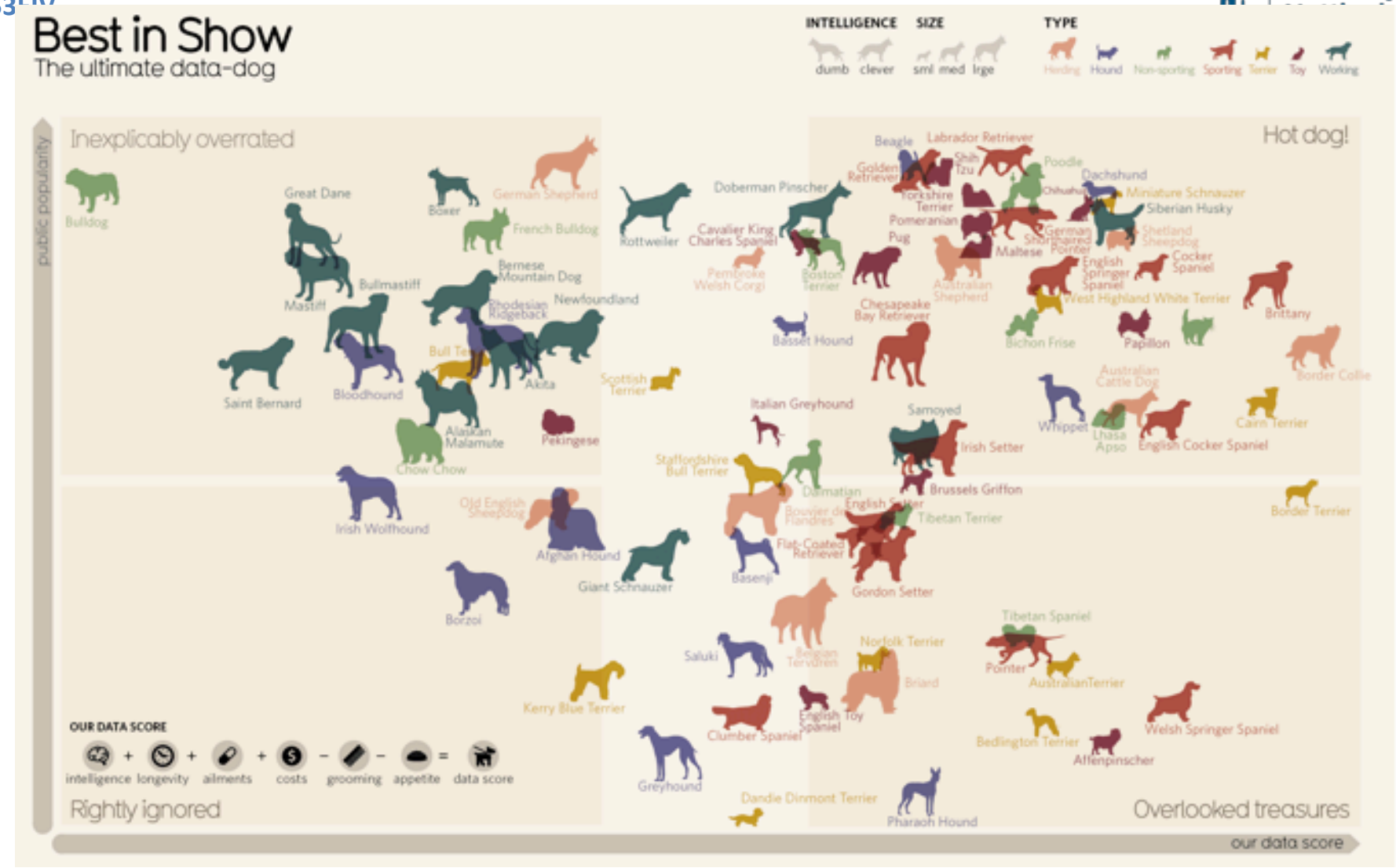
Multivariate Data Visualization

Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

When to use?

- Use tables when
  - The document will be used to look up individual values
  - The document will be used to compare individual values
  - Precise values are required
  - The quantitative info to be communicated involves more than one unit of measure
- Use graphs when
  - The message is contained in the shape of the values
  - The document will be used to reveal relationships among values
  - Especially useful when the number of data points is huge

(Optional Reading) Stephen Few. 2012. Show Me the Numbers: Designing Tables and Graphs to Enlighten (2nd ed.). Analytics Press, , USA.

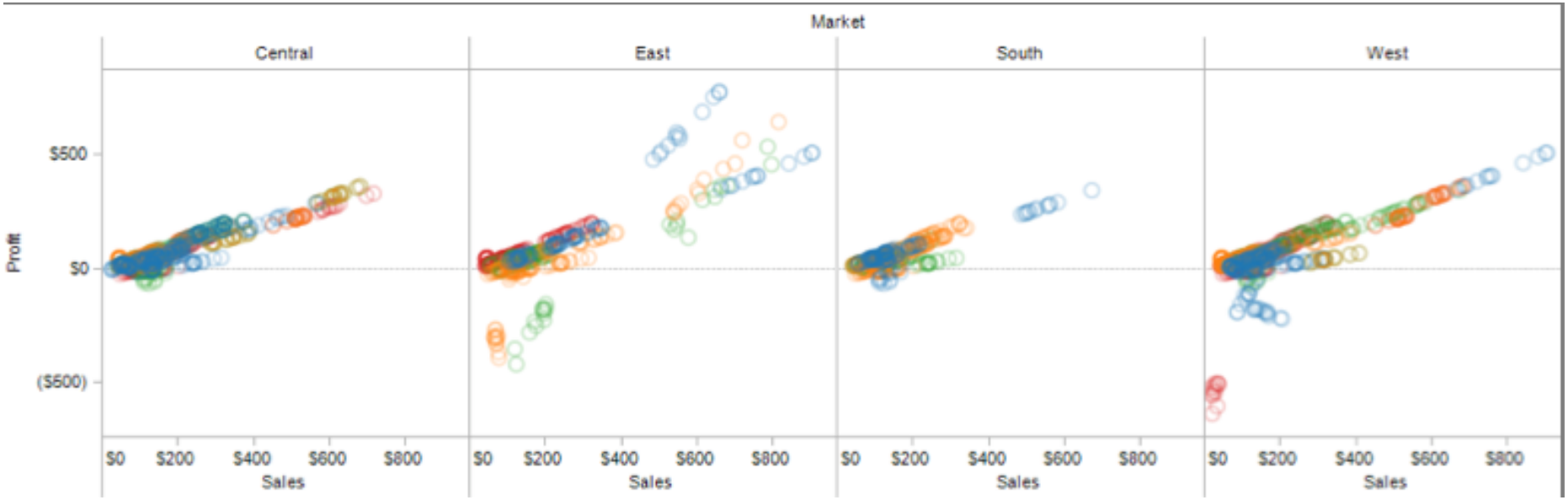


- Iconic Representations: Glyph (graphical object) represents a data case
- Visual properties of glyph represent different variables

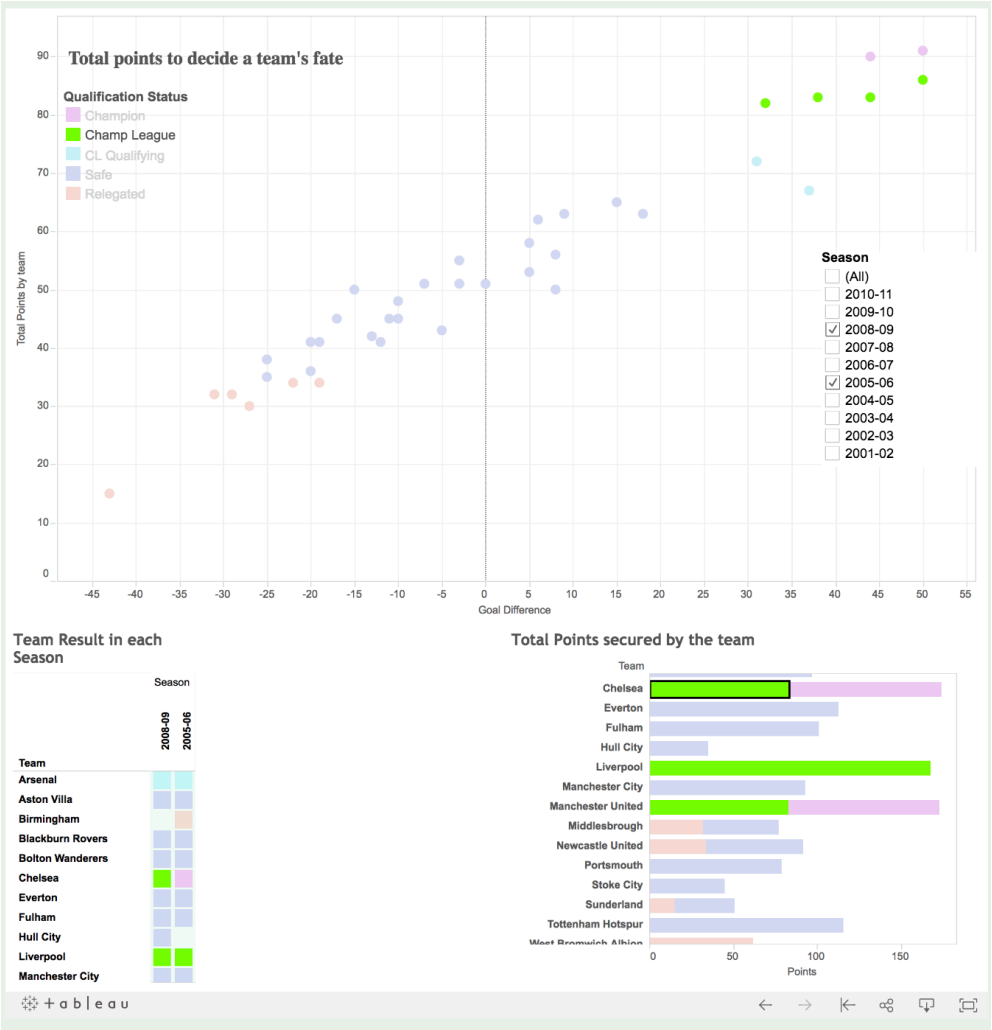
Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

Trellis Display (Small Multiples)

- It subdivides space to enable comparison across multiple plots.
- Typically nominal or ordinal variables are used as dimensions for subdivision.



Multiple Coordinated Views



https://kpotluri.github.io/SoccerGuru/

Minard 1869: Napoleon's March

Y-axis: longitude (Q)

X-axis: latitude (Q)

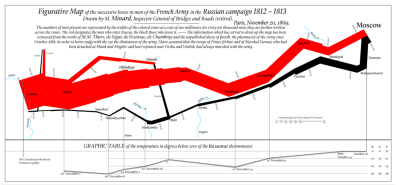
Width: army size (Q)

+

Y-axis: temperature (Q)

X-axis: longitude (Q) / time (O)

=

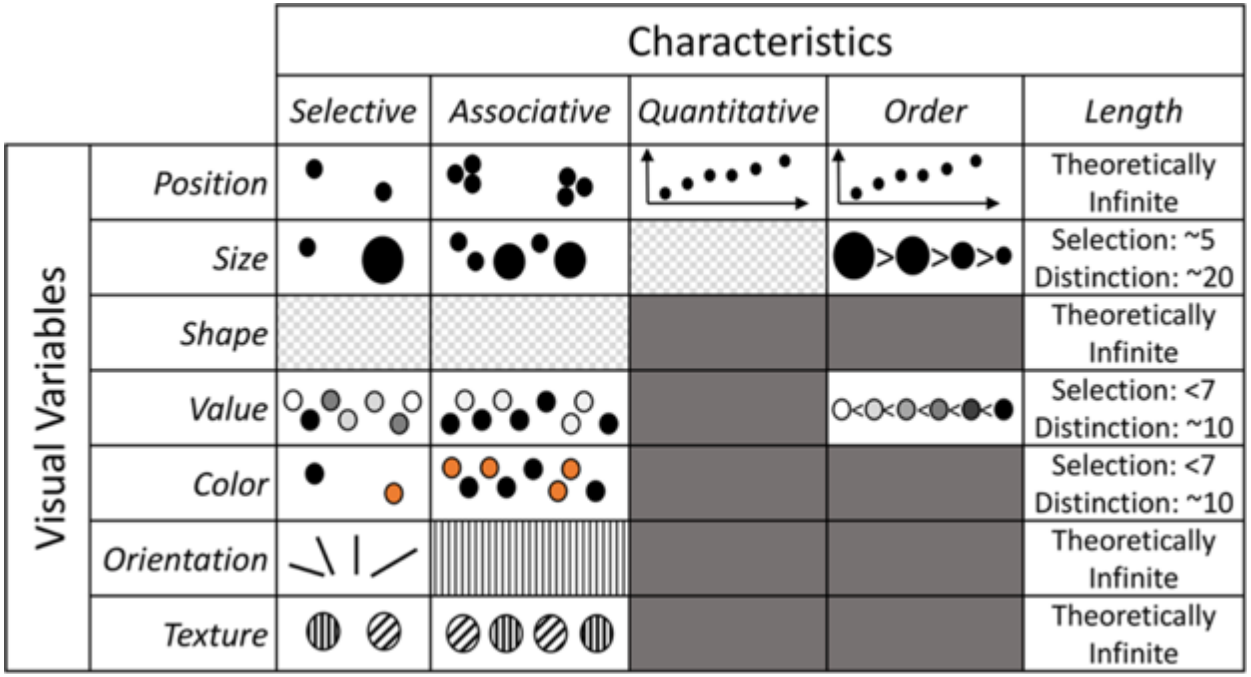


Depicts at least 5 quantitative variables.

Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

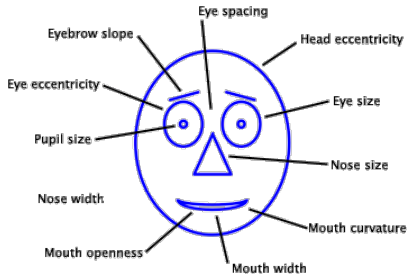
Multivariate Data Visualization

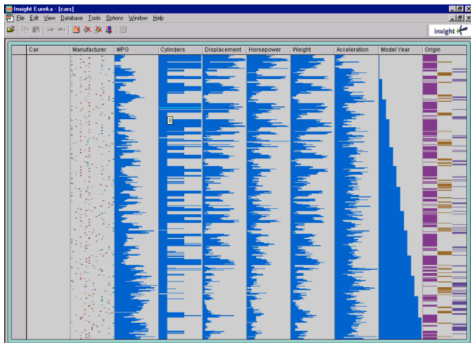
- Visual Encodings: 8 dimensions?
- Focus: techniques can generally handle all data sets

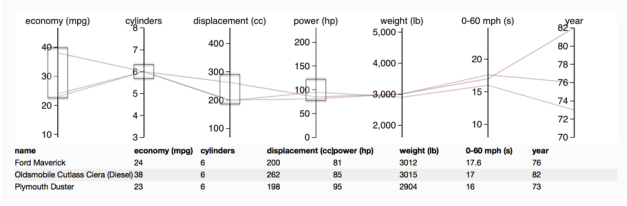


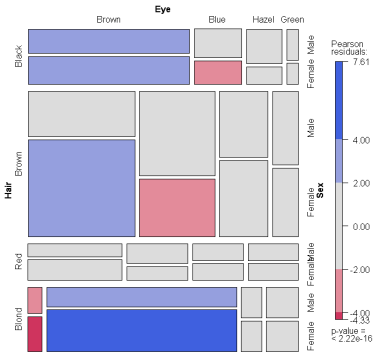
Common Multivariate Data Visualization Techniques

- Chernoff Faces
- Table Lens
- Parallel Coordinates
- Mosaic Plot







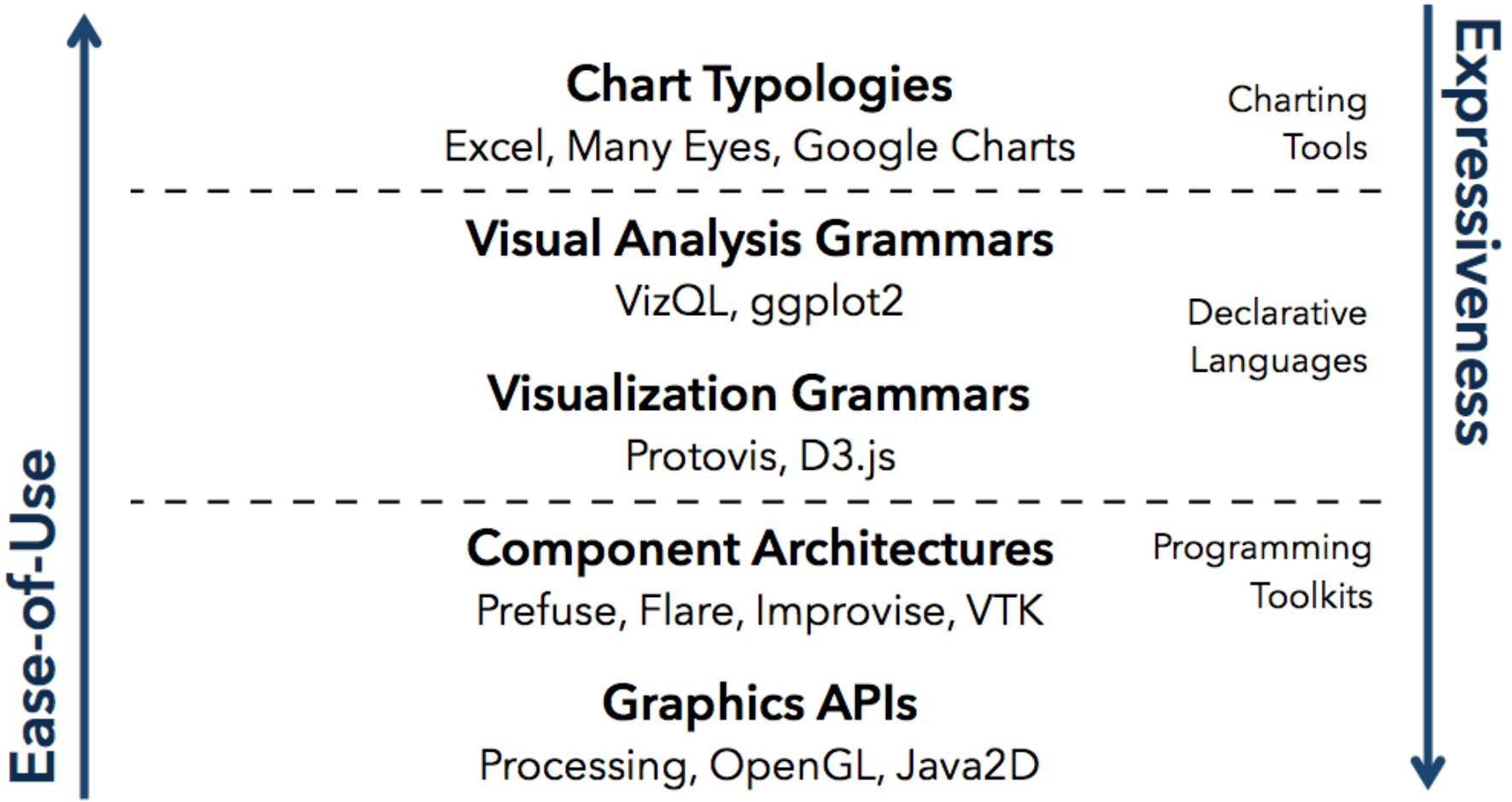


Multivariate Data Visualization

- Strategies:
  - Avoid “over-encoding”
  - Use space and small multiples intelligently
  - Reduce the problem space
  - Use interaction to generate relevant views

- Rarely does a single visualization answer all questions. Instead, the ability to generate appropriate visualizations quickly is key.

Visualization Tools



The Advantages of Declarative Languages

- **Faster iteration**. Less code. Larger user base.
- **Better visualization**. Smart defaults.
- **Reuse**. Write-once, then re-apply.
- **Performance**. Optimization, scalability.
- **Portability**. Multiple devices, renderers, inputs.
- **Programmatic generation**. Write programs which output visualizations. Automated search & recommendation.

Building a Plot in ggplot2

data to visualize (a data frame)

map variables to ***aes***thetic attributes

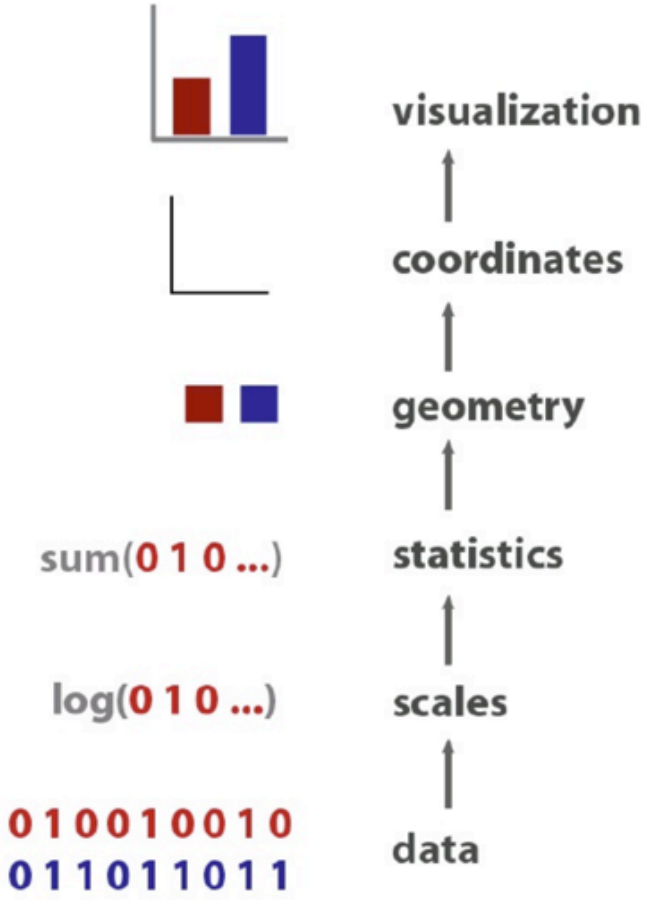
geometric objects – what you see (points, bars, etc)

scales map values from data to aesthetic space

faceting subsets the data to show multiple plots

statistical transformations – summarize data

coordinate systems put data on plane of graphic



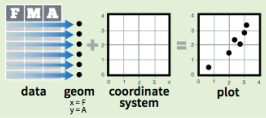
Wickham 2009

Data Visualization with ggplot2 Cheat Sheet

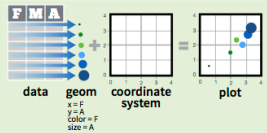
RStudio

Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same few components: a **data** set, a set of **geoms**—visual marks that represent data points, and a **coordinate system.**



To display data values, map variables in the data set to aesthetic properties of the geom like **size**, **color**, and **x** and **y** locations.



Build a graph with **qplot()** or **ggplot()**

aesthetic mappings | data | geom

qplot(x = cty, y = hwy, color = cyl, data = mpg, geom = "point"**)**

Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

ggplot(data = mpg, **aes(**x = cty, y = hwy**))**

Begins a plot that you finish by adding layers to. No defaults, but provides more control than qplot().

data

```
ggplot(mpg, aes(hwy, cty)) +
 geom_point(aes(color = cyl)) +
 geom_smooth(method ="lm") +
 coord_cartesian() +
 scale_color_gradient() +
 theme_bw()
```

add layers, elements with +

layer = geom + default stat + layer specific mappings

additional elements

Add a new layer to a plot with a **geom_*()** or **stat_*()** function. Each provides a geom, a set of aesthetic mappings, and a default stat and position adjustment.

last_plot()

Returns the last plot

ggsave("plot.png", width = 5, height = 5)

Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

One Variable

Continuous

a <- ggplot(mpg, aes(hwy))

a + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size
b + geom_area(aes(y = ..density..), stat = "bin")

a + geom_density(kernel = "gaussian"**)**
x, y, alpha, color, fill, linetype, size, weight
b + geom_density(aes(y = ..county..))

a + geom_dotplot()
x, y, alpha, color, fill

a + geom_freqpoly()
x, y, alpha, color, linetype, size
b + geom_freqpoly(aes(y = ..density..))

a + geom_histogram(binwidth = 5**)**
x, y, alpha, color, fill, linetype, size, weight
b + geom_histogram(aes(y = ..density..))

Discrete

b <- ggplot(mpg, aes(fl))

b + geom_bar()
x, alpha, color, fill, linetype, size, weight

Graphical Primitives

c <- ggplot(map, aes(long, lat))

c + geom_polygon(aes(group = group)**)**
x, y, alpha, color, fill, linetype, size

d <- ggplot(economics, aes(date, unemploy))

d + geom_path(lineend="butt", linejoin="round', linemitre=1**)**
x, y, alpha, color, linetype, size

d + geom_ribbon(aes(ymin=unemploy - 900, ymax=unemploy + 900**)**
x, ymax, ymin, alpha, color, fill, linetype, size

e <- ggplot(seals, aes(x = long, y = lat))

e + geom_segment(aes(
xend = long + delta_long,
yend = lat + delta_lat**)**
x, xend, y, yend, alpha, color, linetype, size

e + geom_rect(aes(xmin = long, ymin = lat, xmax= long + delta_long, ymax = lat + delta_lat**)**
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

Two Variables

Continuous X, Continuous Y

f <- ggplot(mpg, aes(cty, hwy))

f + geom_blank()

f + geom_jitter()
x, y, alpha, color, fill, shape, size

f + geom_point()
x, y, alpha, color, fill, shape, size

f + geom_quantile()
x, y, alpha, color, linetype, size, weight

f + geom_rug(sides = "bl"**)**
alpha, color, linetype, size

f + geom_smooth(model = lm**)**
x, y, alpha, color, fill, linetype, size, weight

f + geom_text(aes(label = cty)**)**
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

Discrete X, Continuous Y

g <- ggplot(mpg, aes(class, hwy))

g + geom_bar(stat = "identity")
x, y, alpha, color, fill, linetype, size, weight

g + geom_boxplot()
lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, shape, size, weight

g + geom_dotplot(binaxis = "y", stackdir = "center"**)**
x, y, alpha, color, fill

g + geom_violin(scale = "area"**)**
x, y, alpha, color, fill, linetype, size, weight

Discrete X, Discrete Y

h <- ggplot(diamonds, aes(cut, color))

h + geom_jitter()
x, y, alpha, color, fill, shape, size

Continuous Bivariate Distribution

i <- ggplot(movies, aes(year, rating))

i + geom_bin2d(binwidth = c(0.5, 0.5)**)**
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight

i + geom_density2d()
x, y, alpha, colour, linetype, size

i + geom_hex()
x, y, alpha, colour, fill size

Continuous Function

j <- ggplot(economics, aes(date, unemploy))

j + geom_area()
x, y, alpha, color, fill, linetype, size

j + geom_line()
x, y, alpha, color, linetype, size

j + geom_step(direction = "hv"**)**
x, y, alpha, color, linetype, size

Visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
k <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))

k + geom_crossbar(fatten = 2**)**
x, y, ymax, ymin, alpha, color, fill, linetype, size

k + geom_errorbar()
x, ymax, ymin, alpha, color, linetype, size, width (also **geom_errorbarh()**)

k + geom_linerange()
x, ymin, ymax, alpha, color, linetype, size

k + geom_pointrange()
x, y, ymin, ymax, alpha, color, fill, linetype, shape, size

Maps

data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))
map <- map_data("state")
l <- ggplot(data, aes(fill = murder))

l + geom_map(aes(map_id = state), map = map**) + expand_limits(**x = map$long, y = map$lat**)**
map_id, alpha, color, fill, linetype, size

Three Variables

seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))
m <- ggplot(seals, aes(long, lat))

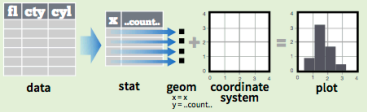
m + geom_contour(aes(z = z)**)**
x, y, z, alpha, colour, linetype, size, weight

m + geom_raster(aes(fill = z), hjust=0.5, vjust=0.5, interpolate=FALSE**)**
x, y, alpha, fill

m + geom_tile(aes(fill = z)**)**
x, y, alpha, color, fill, linetype, size

Stats - An alternative way to build a layer

Some plots visualize a **transformation** of the original data set. Use a **stat** to choose a common transformation to visualize, e.g. **a + geom_bar(stat = "bin")**



Each stat creates additional variables to map aesthetics to. These variables use a common **..name..** syntax.

stat functions and geom functions both combine a stat with a geom to make a layer, i.e. **stat_bin(geom="bar")** does the same as **geom_bar(stat="bin")**

stat function | layer specific mappings | variable created by transformation

i + stat_density2d(aes(fill = ..level..), geom = "polygon", n = 100)

geom for layer | parameters for stat

1D distributions

a + stat_bin(binwidth = 1, origin = 10)
x, y | ..count.., ..ncount.., ..density.., ..ndensity..
a + stat_bindot(binwidth = 1, binaxis = "x")
x, y, | ..count.., ..ncount..
a + stat_density(adjust = 1, kernel = "gaussian")
x, y, | ..count.., ..density.., ..scaled..

2D distributions

f + stat_bin2d(bins = 30, drop = TRUE)
x, y, fill | ..count.., ..density..
f + stat_binhex(bins = 30)
x, y, fill | ..count.., ..density..
f + stat_density2d(contour = TRUE, n = 100)
x, y, color, size | ..level..

3 Variables

m + stat_contour(aes(z = z))
x, y, z, order | ..level..
m + stat_spoke(aes(radius= z, angle = z))
angle, radius, x, xend, y, yend | ..x.., ..xend.., ..y.., ..yend..
m + stat_summary_hex(aes(z = z), bins = 30, fun = mean)
x, y, z, fill | ..value..
m + stat_summary2d(aes(z = z), bins = 30, fun = mean)
x, y, z, fill | ..value..

Comparisons

g + stat_boxplot(coef = 1.5)
x, y | ..lower.., ..middle.., ..upper.., ..outliers..
g + stat_ydensity(adjust = 1, kernel = "gaussian", scale = "area")
x, y | ..density.., ..scaled.., ..count.., ..n.., ..violinwidth.., ..width..

Functions

f + stat_ecdf(n = 40)
x, y | ..x.., ..y..
f + stat_quantile(quantiles = c(0.25, 0.5, 0.75), formula = y ~ log(x), method = "rq")
x, y | ..quantile.., ..x.., ..y..
f + stat_smooth(method = "auto", formula = y ~ x, se = TRUE, n = 80, fullrange = FALSE, level = 0.95)
x, y | ..se.., ..x.., ..y.., ..ymin.., ..ymax..

General Purpose

ggplot() + stat_function(aes(x = -3:3), fun = dnorm, n = 101, args = list(sd=0.5))
x | ..y..
f + stat_identity()
ggplot() + stat_qq(aes(sample=1:100), distribution = qt, dparams = list(df=5))
sample, x, y | ..x.., ..y..
f + stat_sum()
x, y, size | ..size..
f + stat_summary(fun.data = "mean_cl_boot")
f + stat_unique()

Scales

Scales control how a plot maps data values to the visual values of an aesthetic. To change the mapping, add a custom scale.

n <- b + geom_bar(aes(fill = fl))
n

scale_ | aesthetic to adjust | prepackaged scale to use | scale specific arguments

n + scale_fill_manual(
values = c("skyblue", "royalblue", "blue", "navy"),
limits = c("d", "e", "p", "r"), breaks =c("d", "e", "p", "r"),
name = "fuel", labels = c("D", "E", "P", "R"))

range of values to include in mapping | title to use in legend/axis | labels to use in legend/axis | breaks to use in legend/axis

General Purpose scales

Use with any aesthetic:
alpha, color, fill, linetype, shape, size

scale_*_continuous() - map cont' values to visual values
scale_*_discrete() - map discrete values to visual values
scale_*_identity() - use data values **as** visual values
scale_*_manual(values = c()) - map discrete values to manually chosen visual values

X and Y location scales

Use with x or y aesthetics (x shown here)

scale_x_date(labels = date_format("%m/%d"), breaks = date_breaks("2 weeks")) - treat x values as dates. See ?strptime for label formats.
scale_x_datetime() - treat x values as date times. Use same arguments as scale_x_date().
scale_x_log10() - Plot x on log10 scale
scale_x_reverse() - Reverse direction of x axis
scale_x_sqrt() - Plot x on square root scale

Color and fill scales

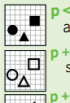
Discrete

n <- b + geom_bar(aes(fill = fl))
n + scale_fill_brewer(palette = "Blues")
For palette choices: library(RColorBrewer) display.brewer.all()
n + scale_fill_grey(start = 0.2, end = 0.8, na.value = "red")

Continuous

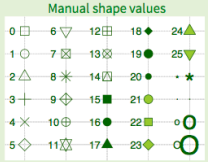
o <- a + geom_dotplot(aes(fill = ..x..))
o + scale_fill_gradient(low = "red", high = "yellow")
o + scale_fill_gradient2(low = "red", hight = "blue", mid = "white", midpoint = 25)
o + scale_fill_gradientn(colours = terrain.colors(6))
Also: rainbow(), heat.colors(), topo.colors(), cm.colors(), RColorBrewer::brewer.pal()

Shape scales



p <- f + geom_point(aes(shape = fl))
p + scale_shape(solid = FALSE)
p + scale_shape_manual(values = c(3:7))
Shape values shown in chart on right

Manual shape values



Size scales

q <- f + geom_point(aes(size = cyl))

q + scale_size_area(max = 6)
Value mapped to area of circle (not radius)

Coordinate Systems

r <- b + geom_bar()

r + coord_cartesian(xlim = c(0, 5))
xlim, ylim
The default cartesian coordinate system

r + coord_fixed(ratio = 1/2)
ratio, xlim, ylim
Cartesian coordinates with fixed aspect ratio between x and y units

r + coord_flip()
xlim, ylim
Flipped Cartesian coordinates

r + coord_polar(theta = "x", direction=1)
theta, start, direction
Polar coordinates

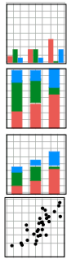
r + coord_trans(xtrans = "sqrt")
xtrans, ytrans, limx, limy
Transformed cartesian coordinates. Set extras and strains to the name of a window function.

z + coord_map(projection = "ortho", orientation=c(41, -74, 0))
projection, orientation, xlim, ylim
Map projections from the mapproj package (mercator (default), azequalarea, lagrange, etc.)

Position Adjustments

Position adjustments determine how to arrange geoms that would otherwise occupy the same space.

s <- ggplot(mpg, aes(fl, fill = drv))



s + geom_bar(position = "dodge")
Arrange elements side by side

s + geom_bar(position = "fill")
Stack elements on top of one another, normalize height

s + geom_bar(position = "stack")
Stack elements on top of one another

f + geom_point(position = "jitter")
Add random noise to X and Y position of each element to avoid overplotting

Each position adjustment can be recast as a function with manual **width** and **height** arguments

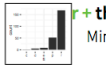
s + geom_bar(position = position_dodge(width = 1))

Themes

r + theme_bw()
White background with grid lines

r + theme_classic()
White background no gridlines

r + theme_grey()
Grey background (default theme)



r + theme_minimal()
Minimal theme

ggthemes - Package with additional ggplot2 themes

Faceting

Facets divide a plot into subplots based on the values of one or more discrete variables.

t <- ggplot(mpg, aes(cty, hwy)) + geom_point()

t + facet_grid(. ~ fl)
facet into columns based on fl

t + facet_grid(year ~ .)
facet into rows based on year

t + facet_grid(year ~ fl)
facet into both rows and columns

t + facet_wrap(~ fl)
wrap facets into a rectangular layout

Set **scales** to let axis limits vary across facets

t + facet_grid(y ~ x, scales = "free")
x and y axis limits adjust to individual facets
- **"free_x"** - x axis limits adjust
- **"free_y"** - y axis limits adjust

Set **labeller** to adjust facet labels

t + facet_grid(. ~ fl, labeller = label_both)

fl: c	fl: d	fl: e	fl: p	fl: r

t + facet_grid(. ~ fl, labeller = label_bquote(alpha ^ .(x)))**

α^{c} α^{d} α^{e} α^{p} α^{r}

t + facet_grid(. ~ fl, labeller = label_parsed)

c	d	e	p	r

Labels

t + ggtitle("New Plot Title")
Add a main title above the plot
t + xlab("New X label")
Change the label on the X axis
t + ylab("New Y label")
Change the label on the Y axis
t + labs(title =" New title", x = "New x", y = "New y")
All of the above

Use scale functions to update legend labels

Legends

t + theme(legend.position = "bottom")
Place legend at "bottom", "top", "left", or "right"
t + guides(color = "none")
Set legend type for each aesthetic: colorbar, legend, or none (no legend)
t + scale_fill_discrete(name = "Title", labels = c("A", "B", "C"))
Set legend title and labels with a scale function.

Zooming

Without clipping (preferred)
t + coord_cartesian(xlim = c(0, 100), ylim = c(10, 20))

With clipping (removes unseen data points)
t + xlim(0, 100) + **ylim(**10, 20)
t + scale_x_continuous(limits = c(0, 100)) + **scale_y_continuous(**limits = c(0, 100))

Visualizing Text

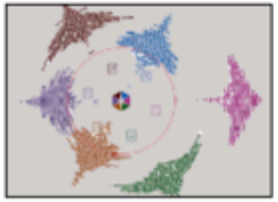
Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

Challenges

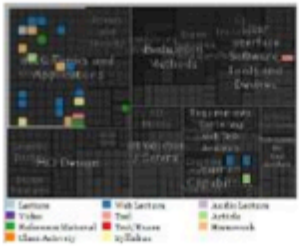
- High Dimensionality
  - Where possible use text to represent text...
  - ... which terms are the most descriptive?
- Context and Semantics
  - Provide relevant context to aid understanding.
  - Show (or provide access to) the source text.
- Modeling Abstraction
  - Determine your analysis task.
  - Understand abstraction of your language models.
  - Match analysis task with appropriate tools and models.

Text Visualization

Visualizing text
Showing words, phrases, and sentences



Visualization for IR
Helping search



- Content
- Context
- Relationship to others

Text Processing Pipeline

- Tokenization
  - Segment text into terms.
  - Remove stop words? a, an, the, of, to, be
  - Numbers and symbols? #gocard, @nottinghamforestfball
  - Entities? Nottingham, Trump.
- Stemming
  - Group together different forms of a word.
  - Porter stemmer? visualization(s), visualize(s), visually -> visual
  - Lemmatization? goes, went, gone -> go
- Ordered list of terms

Bag of Words Model

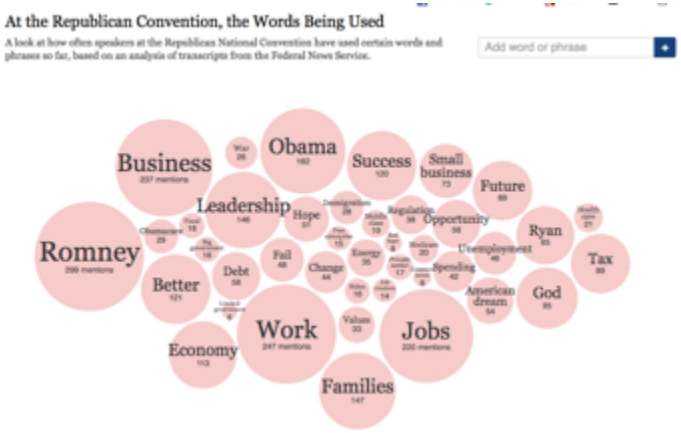
- Ignore ordering relationships within the text
- A document ≈ vector of term weights
  - Each dimension corresponds to a term (10,000+)
  - Each value represents the relevance
    - For example, simple term counts
- Aggregate into a document-term matrix
  - Document vector space model

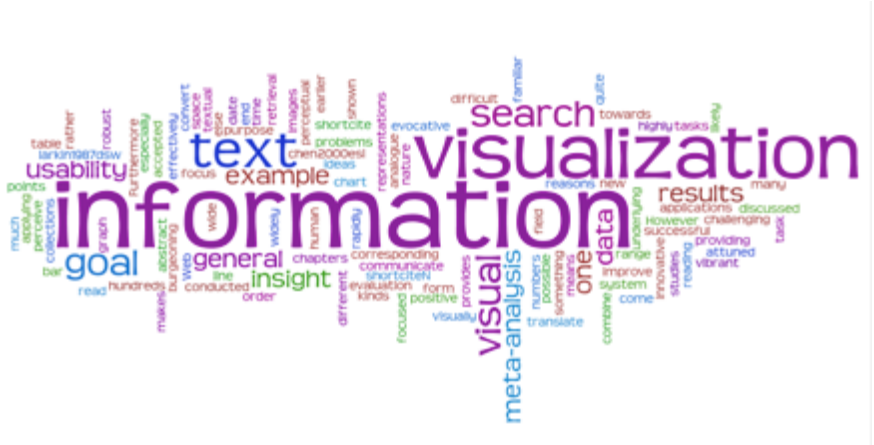
Keyword Weighting

- Term Frequency
  - tf_{td} = count(t) in d
  - Can take log frequency: $\log(1 + tf_{td})$
  - Can normalize to show proportion ($tf_{td} / \Sigma_t\ tf_{td}$)

- TF.IDF: Term Freq by Inverse Document Freq
  - $tf.idf_{td} = \log(1 + tf_{td}) \times \log(N/df_t)$
    - df_t = # docs containing t;
    - N = # of docs

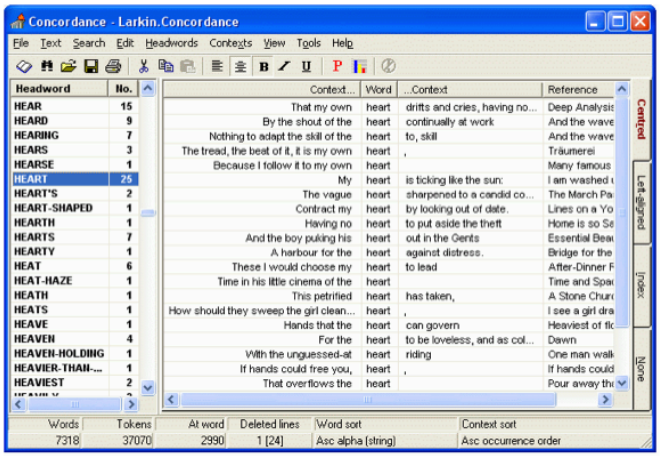
Word Counts and Tag Cloud

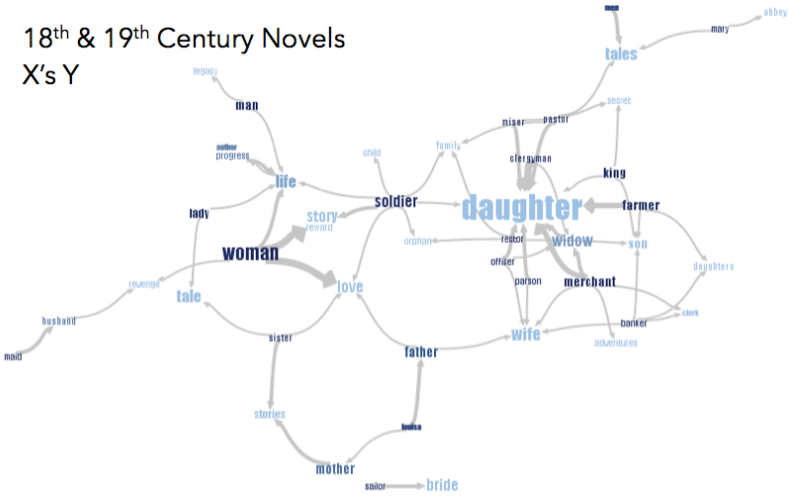




http://www.nytimes.com/interactive/2012/08/28/us/politics/convention-word-counts.html

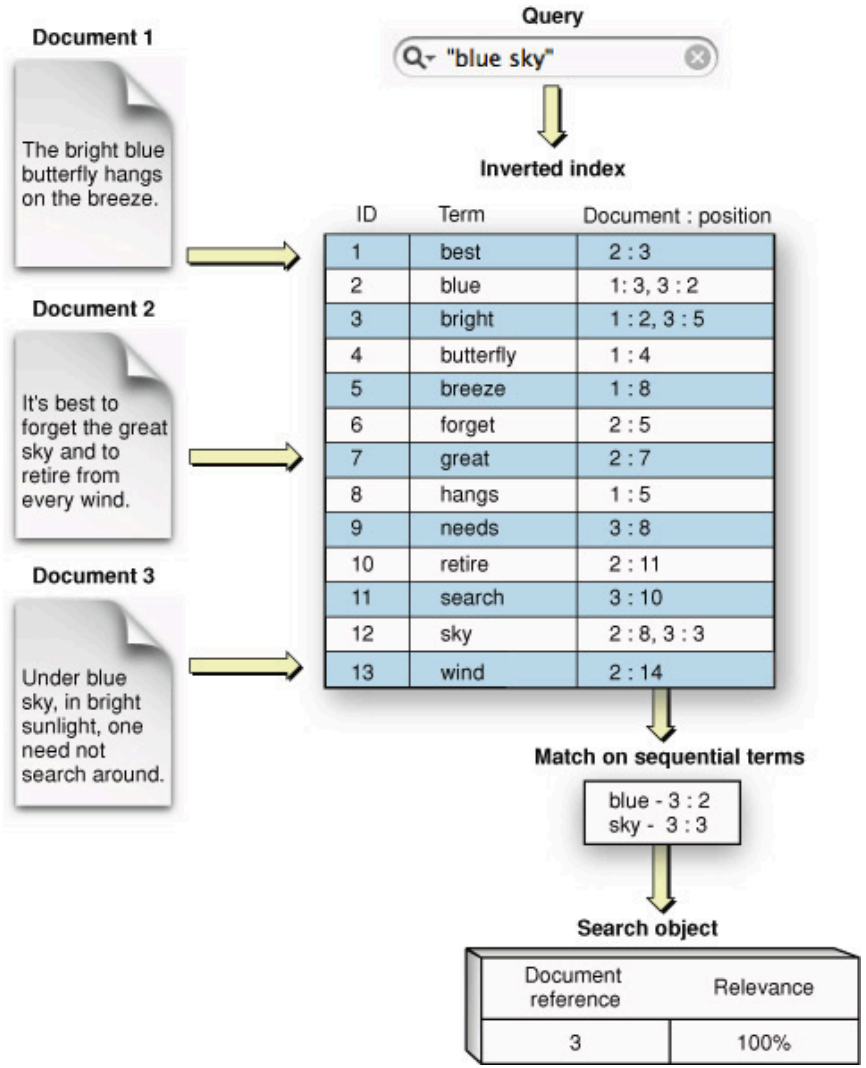
Context and Semantics





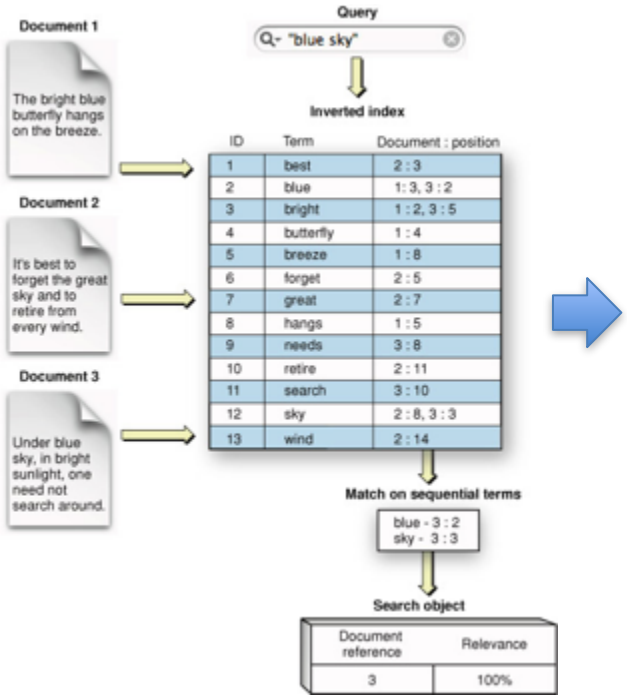
Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

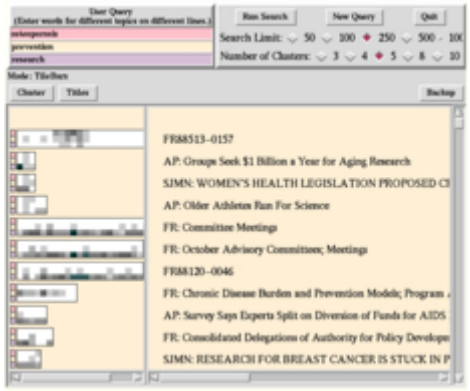
Information Retrieval (IR)

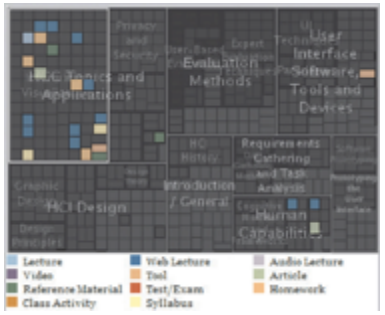


Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

Visualizing for IR







Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

Visualizing Time Series Data

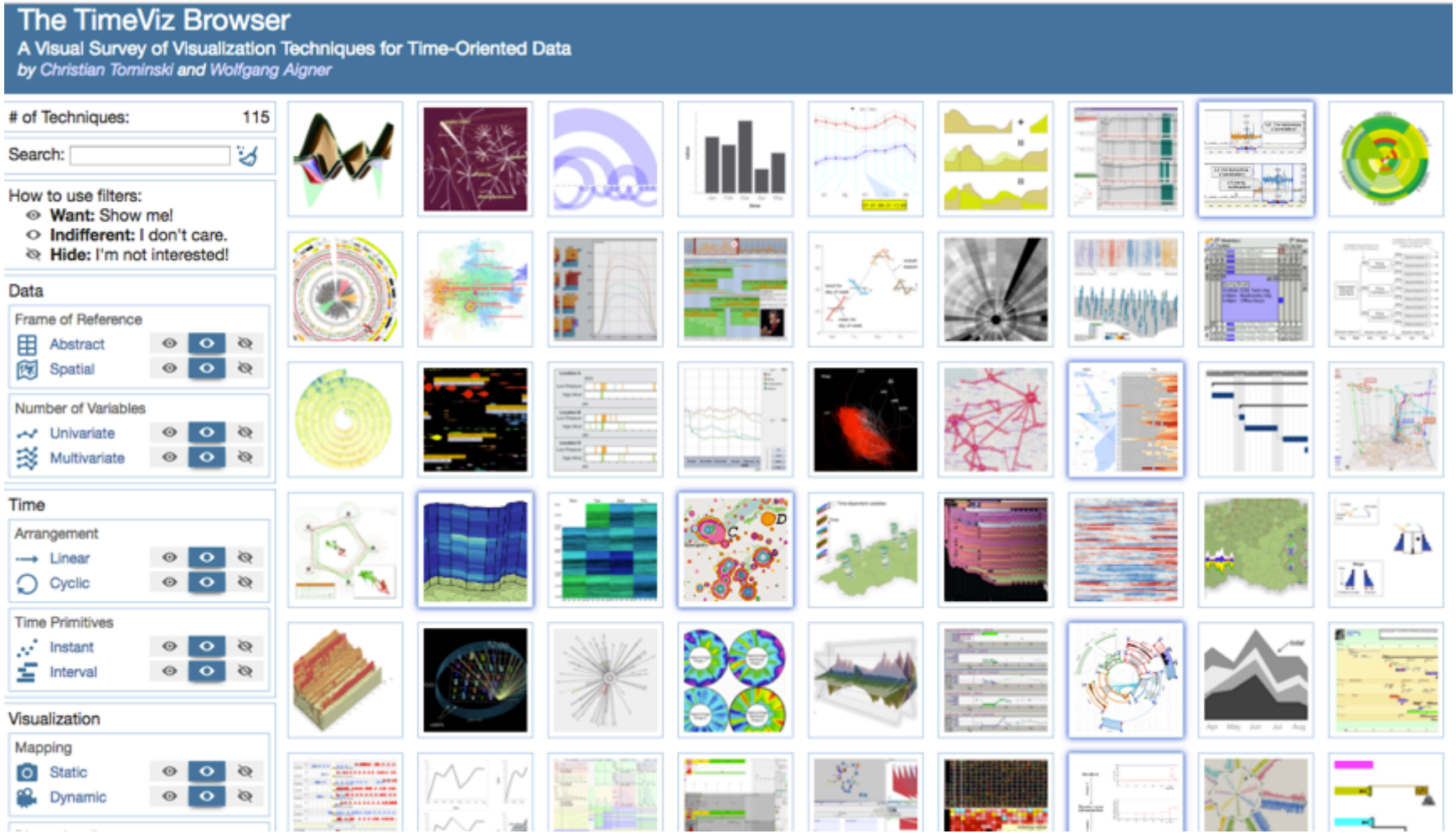
Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

Tasks

- Often asked questions:
  - when was something greatest/least?
  - is there a pattern? are two series similar?
  - does a data element exist at time t, and when?
  - how long does a data element exist and how often?
  - how fast are data elements changing
  - in what order do data elements appear?
  - do data elements exist together?

(Optional Reading) Müller, Wolfgang, and Heidrun Schumann. "Visualization for modeling and simulation: visualization methods for time-dependent data-an overview." Proceedings of the 35th conference on Winter simulation: driving innovation. Winter Simulation Conference, 2003.

Visualizing Time-oriented Data
A Systematic View



http://www.timeviz.net/

Aigner, Wolfgang, et al. "Visualizing time-oriented data—a systematic view." Computers & Graphics 31.3 (2007): 401-409.

Taxonomy

Time	**Temporal primitives**	time points (a) (b) (c) (d) (e) (f) (g) (i)		time intervals (g) (h)
	Structure of time	linear (a) (b) (c) (d) (f) (g) (h) (i)	cyclic (e)	branching (h)
Data	**Frame of reference**	abstract (c) (d) (f) (g) (h) (i)		spatial (a) (b) (e) (i)
	Number of variables	univariate (a) (b) (f) (g) (h)		multivariate (c) (d) (e) (i)
	Level of abstraction	data (a) (b) (c) (d) (e) (f) (g) (h) (i)		data abstractions (b) (g) (i)
Representation	**Time dependency**	static (c) (d) (e) (g) (h) (i)		dynamic (a) (b) (f) (i)
	Dimensionality	2D (a) (c) (d) (g) (h) (i)		3D (b) (e) (f) (i)

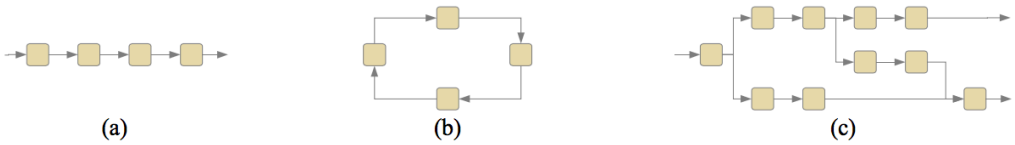
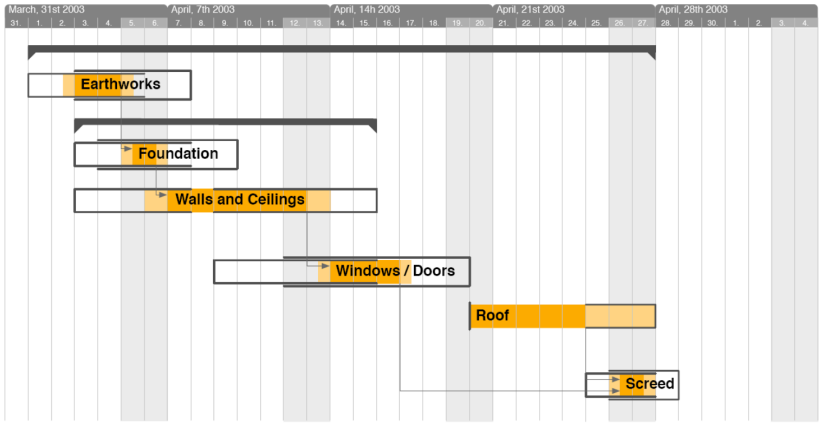
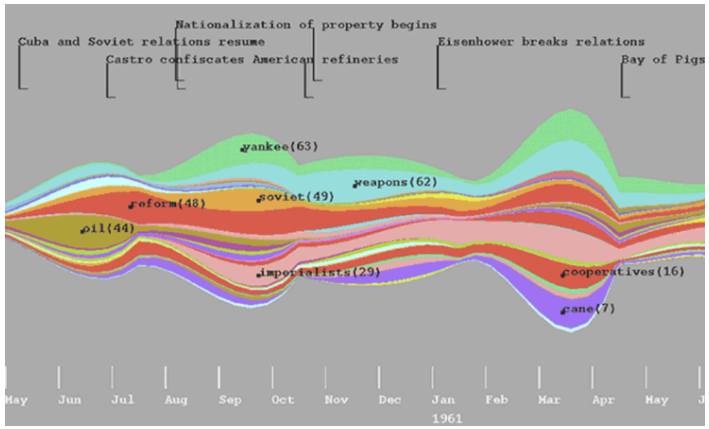


Fig. 2. Structure of time: (a) Linear time; (b) Cyclic time; (c) Branching time.

Aigner, Wolfgang, et al. "Visualizing time-oriented data - a systematic view." Computers & Graphics 31.3 (2007): 401-409.

Time Series Visualization





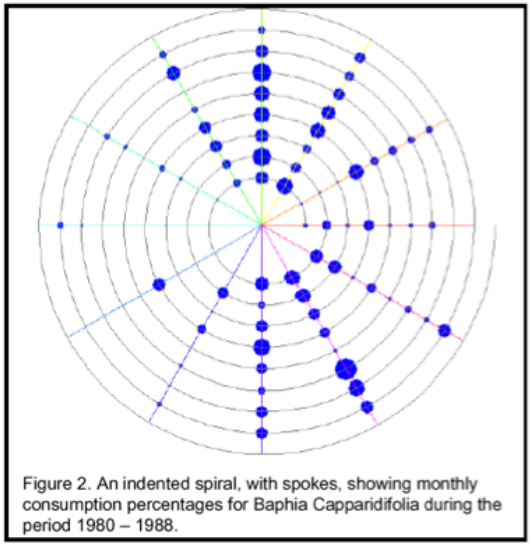
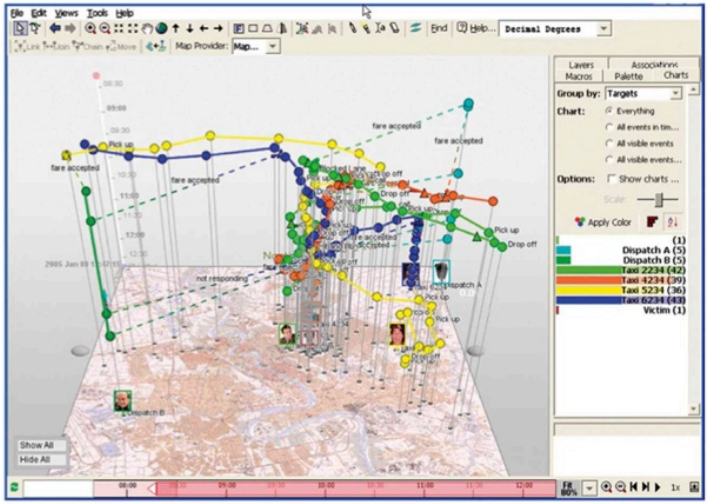


Figure 2. An indented spiral, with spokes, showing monthly consumption percentages for Baphia Capparidifolia during the period 1980 – 1988.



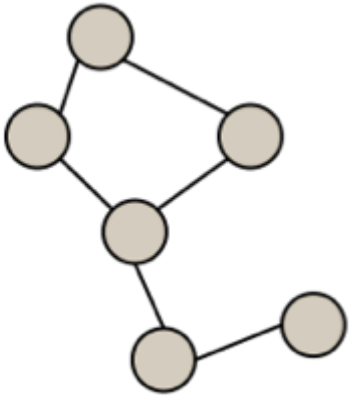
Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

Visualizing Trees and Graphs

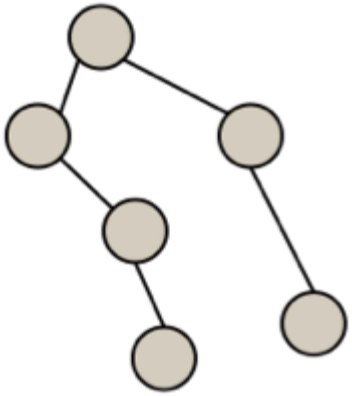
Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

Graphs and Trees

- Graphs
  - Model relations among data
  - Nodes and edges

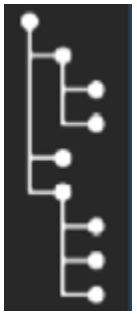


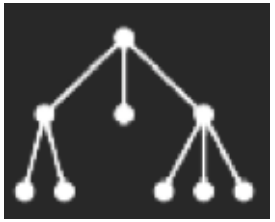
- Trees
  - Graphs with hierarchical structure
    - Connected graph with N-1 edges
  - Nodes as parents and children

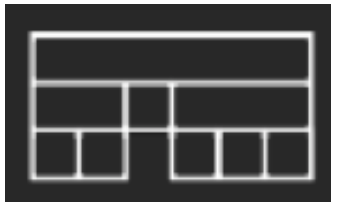


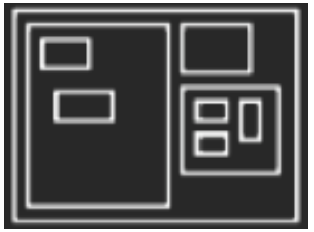
Tree Visualizations

- Indented lists
  - Linear list, indentation encodes depth
- Node-link trees
  - Nodes connected by lines/curves
- Layered diagrams
  - Relative position and alignment
- Treemaps
  - Represent hierarchy by enclosure







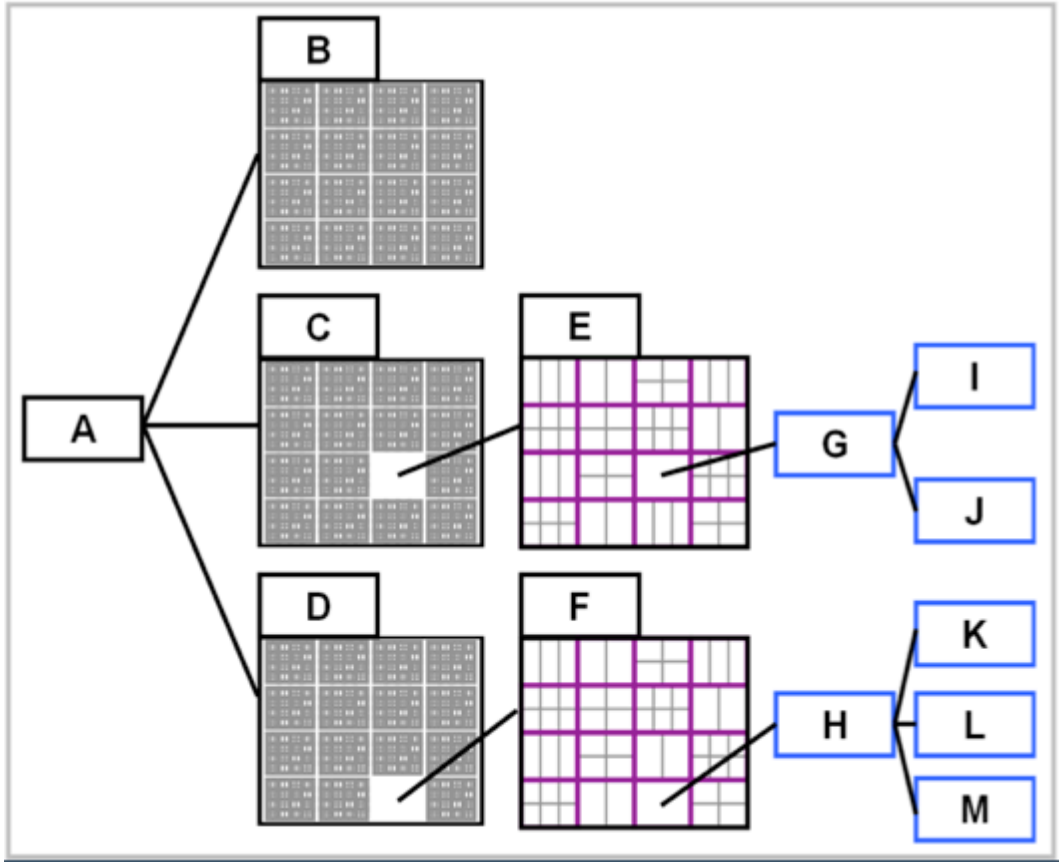


Hybrids

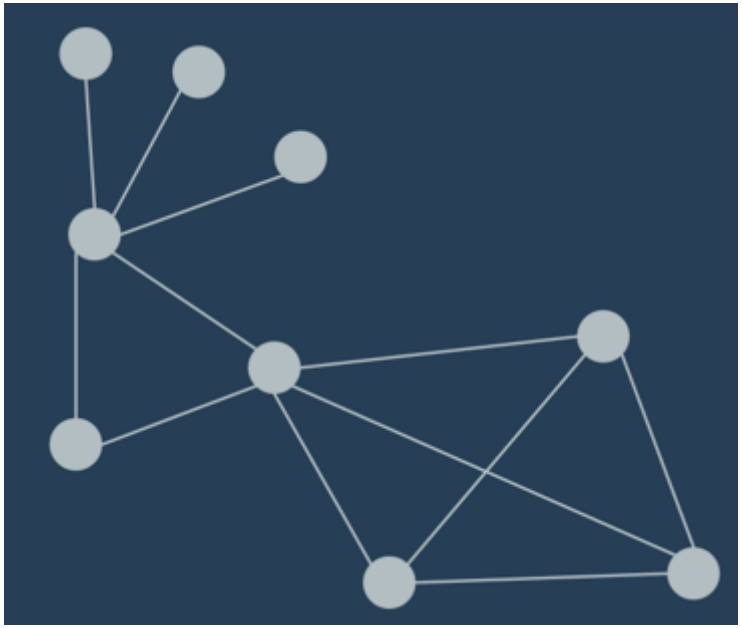
- Elastic Hierarchies
  - Node-link diagram with treemap nodes.

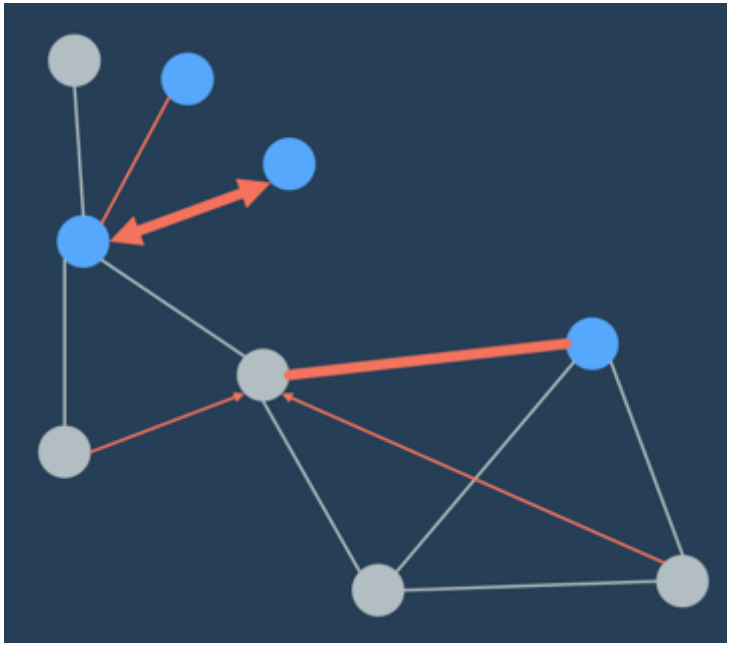
- Video:

https://www.youtube.com/watch?v=nvsIqYQ75yA



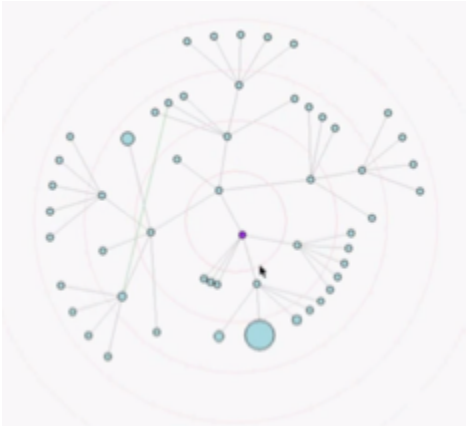
What's in a Graph?





Graph Visualization

- Two representations:
  - Node-link diagrams
  - Matrices



Tree in the Graph

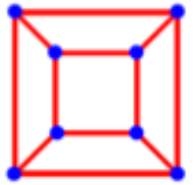
- Major Node-Link Layouts

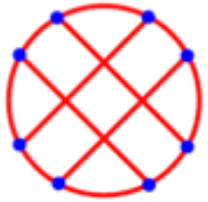
Hierarchical Graph Layout

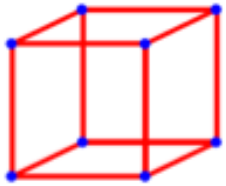
Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

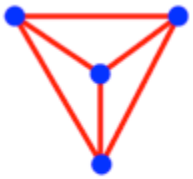
Optimization Techniques

- Treat layout as an optimization problem
  - Define layout using an energy model along with constraints: equations the layout should obey.
  - Use optimization algorithms to solve
- Commonly posed as a physical system
  - Charged particles, springs, drag force, ...
- Different constraints can be introduced
  - Minimize edge crossings
  - Minimize area
  - Minimize line bends
  - Minimize line slopes
  - Maximize smallest angle between edges
  - Maximize symmetry





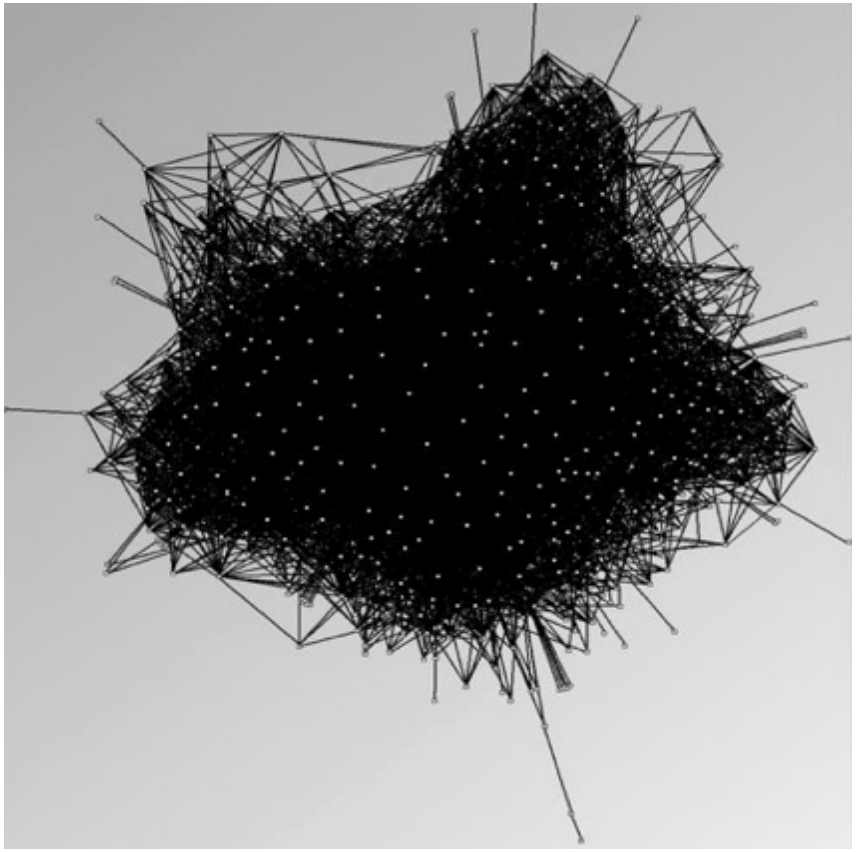


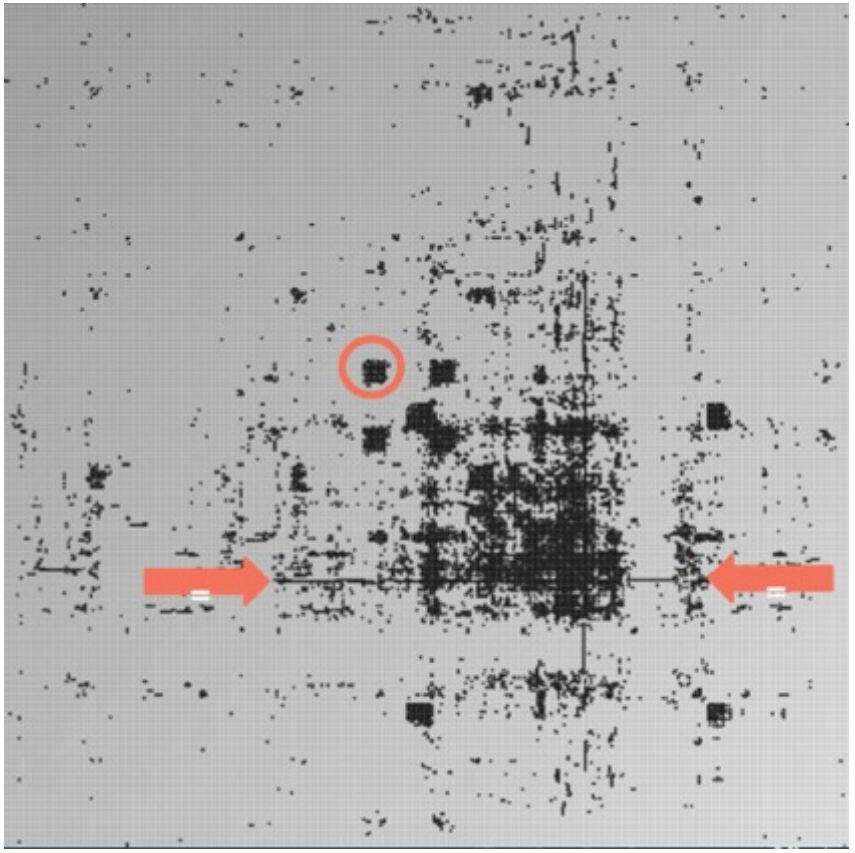


min # crossings

max symmetries

Matrices

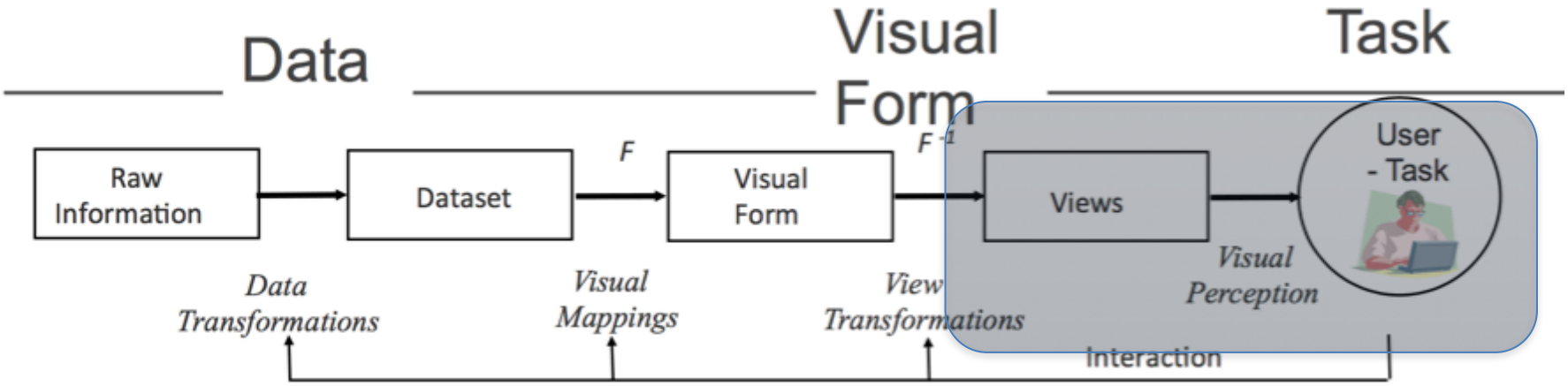


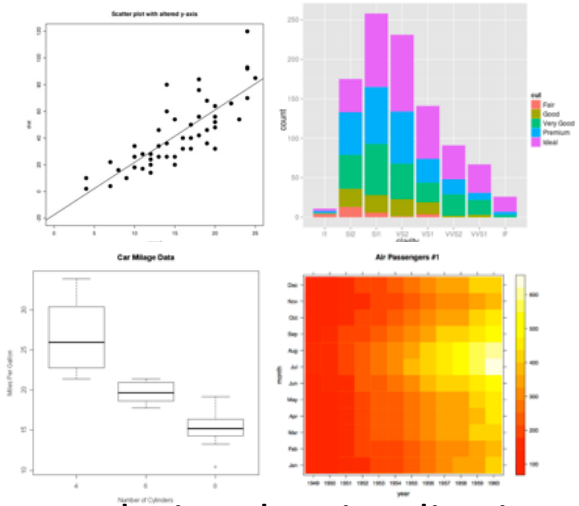


Matrix vs. Node-link

Matrix	Node-link
Require learning	**Familiar**
No overlap	Node overlap
No crossings	Link crossing
Use a lot of space	**More compact**
Dense graphs	**Sparse graphs**

Information Visualization





Design: how can we design the visualization best for human to accomplish their tasks?

What design criteria should we follow?

What Design Criteria to Follow?

- **Expressiveness**
  - A set of facts is expressible in a visual language if the sentences (i.e. the visualizations) in the language (1) express all the facts in the set of data, and (2) only the facts in the data.

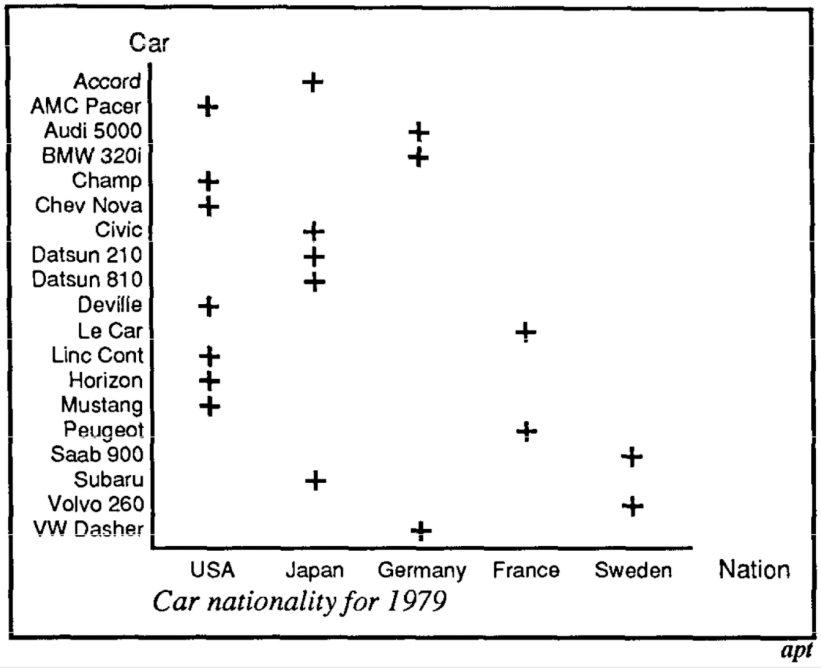
Tell the truth

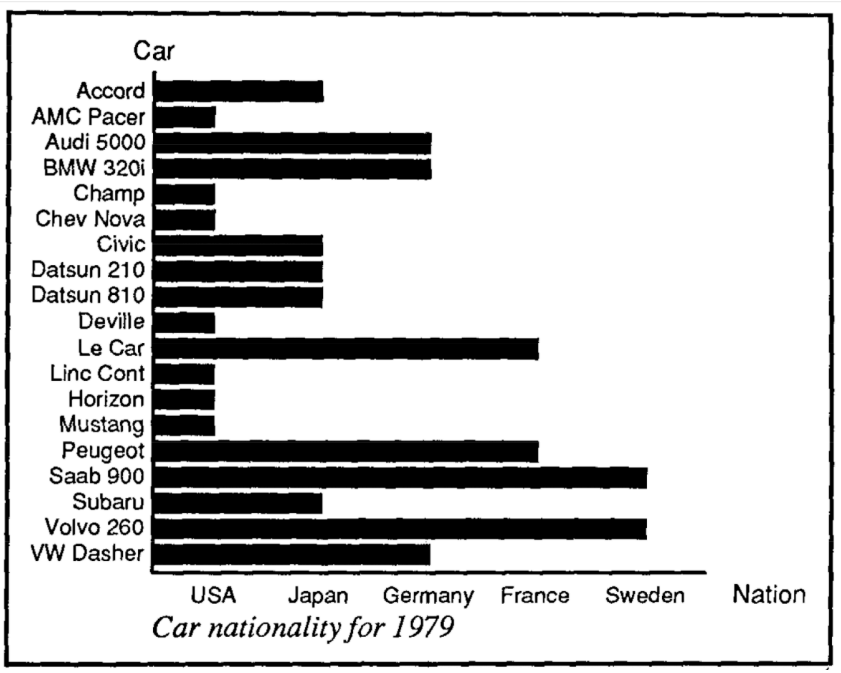
- **Effectiveness**
  - A visualization is more effective than another visualization if the information conveyed by one visualization is more readily perceived than the information in the other visualization.

Use proper encoding

Mackinlay, Automating the design of graphical presentations of relational information, 1986.

Expressiveness





An Alternative

What Design Criteria to Follow?

- **Expressiveness**
  - A set of facts is expressible in a visual language if the sentences (i.e. the visualizations) in the language express (1) all the facts in the set of data, and (2) only the facts in the data.

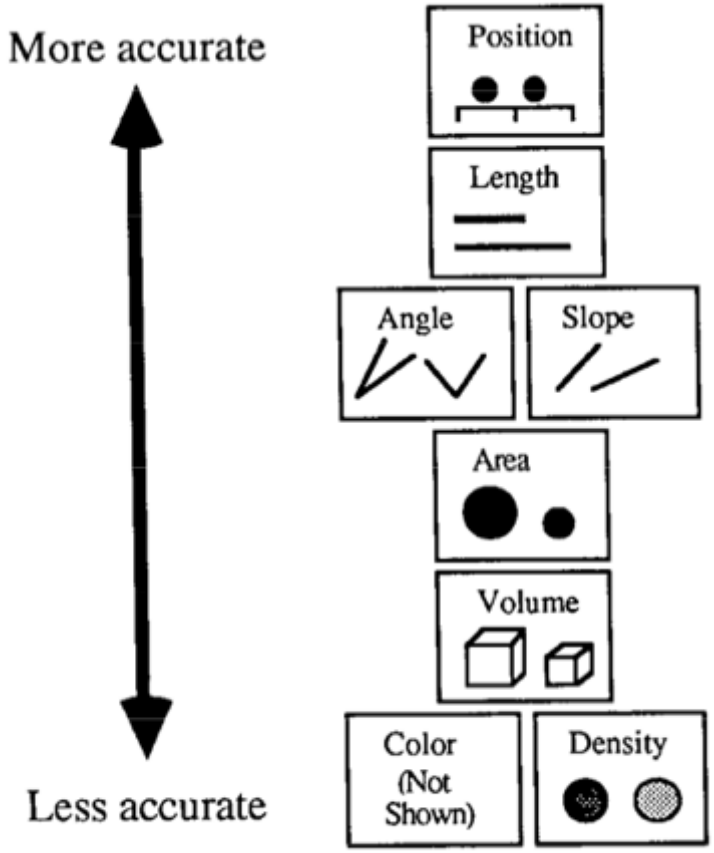
Tell the truth

- **Effectiveness**
  - A visualization is more effective than another visualization if the information conveyed by one visualization is more readily perceived than the information in the other visualization.

Use proper encoding

Mackinlay, Automating the design of graphical presentations of relational information, 1986.

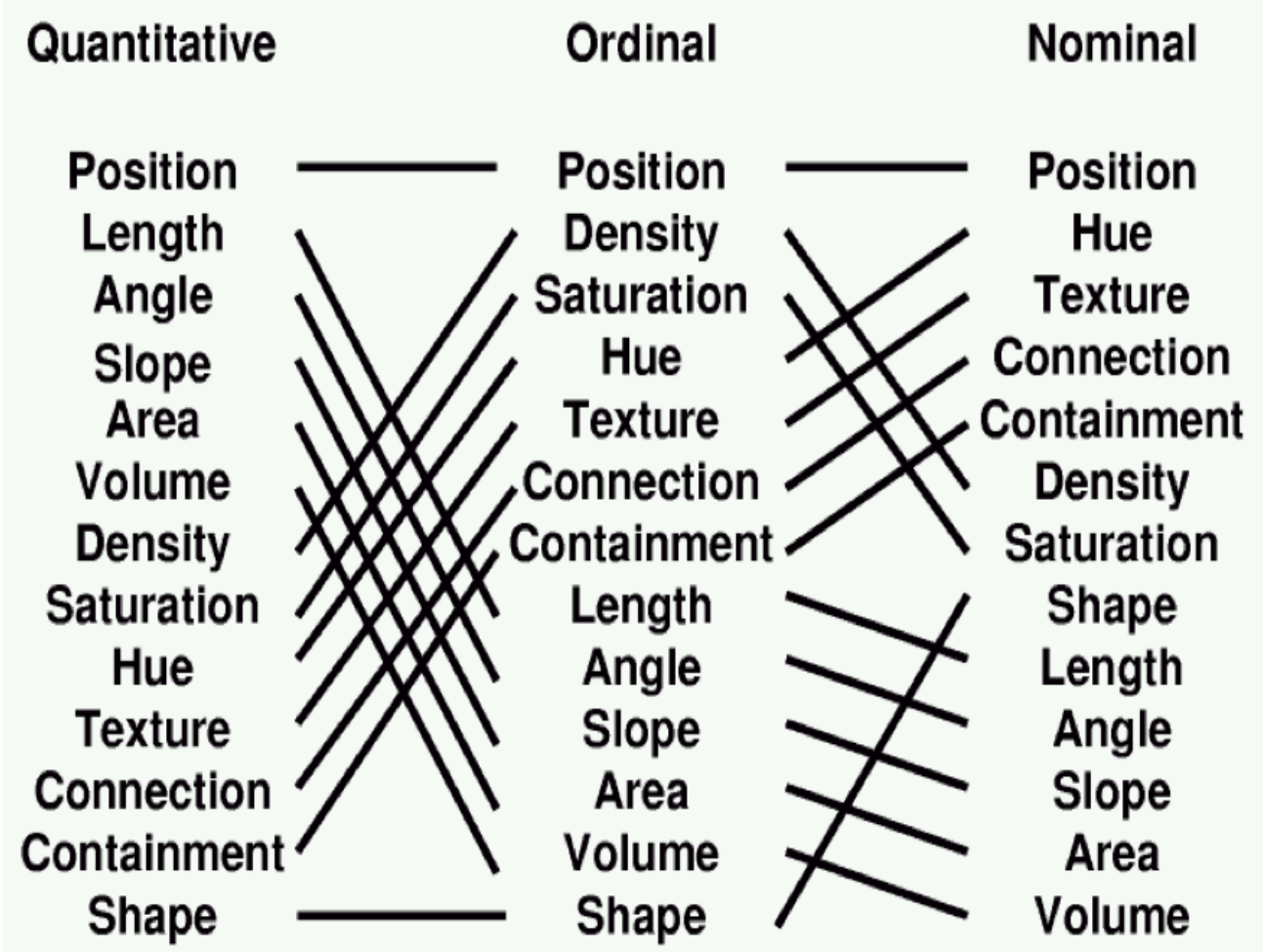
Effectiveness: Accuracy Ranking for Quantitative Information



Mackinlay, Automating the design of graphical presentations of relational information, 1986.

Conjectured Effectiveness of Encodings by Data Type

- Nominal/ Ordinal variables: detect differences
- Quantitative variables: estimate magnitudes



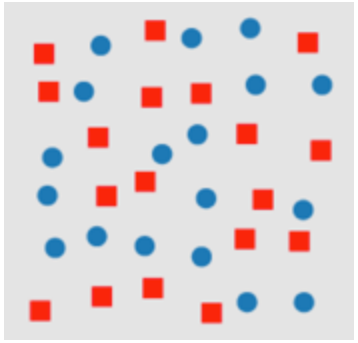
Mackinlay, Automating the design of graphical presentations of relational information, 1986.

Visual Perception

Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

Perceptual Processing Model

- Two stage process
  - Parallel extraction of low-level properties of scene
  - Sequential goal-directed processing



Stage 1

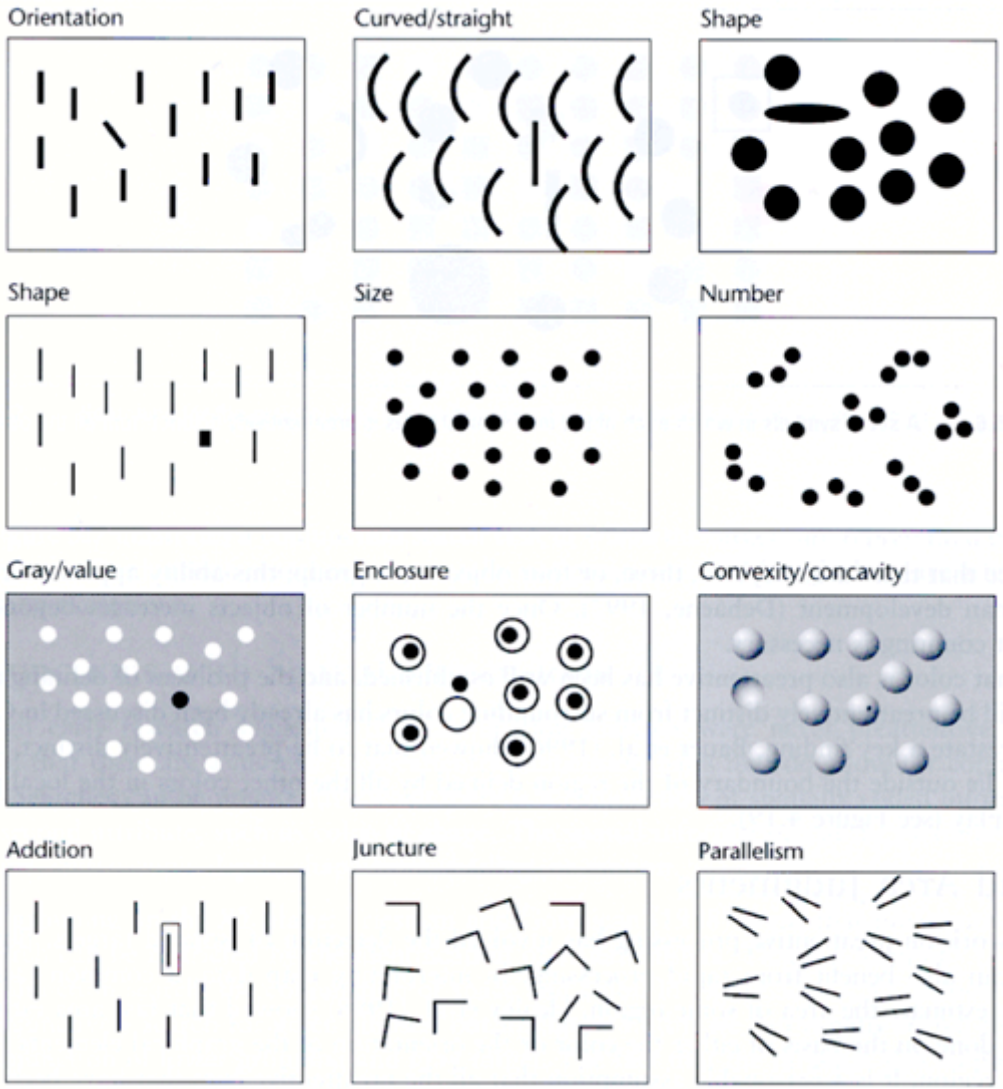
Early, parallel detection of color, texture, shape, spatial attributes

Stage 2

Serial processing of object identification (using memory) and spatial layout, action

Information visualization: Perception for Design (3rd Edition), Colin Ware

Pre-Attentive Features



- length
- width
- size
- curvature
- number
- terminators
- intersection
- closure
- hue
- intensity
- flicker
- direction of motion
- binocular lustre
- stereoscopic depth
- 3-D depth cues
- lighting direction

Information visualization: Perception for Design (3rd Edition), Colin Ware

Pre-Attentive Feature Conjunctions

- Spatial conjunctions are often pre-attentive
- Motion and 3D disparity
- Motion and color
- Motion and shape
- 3D disparity and color
- 3D disparity and shape

- Most conjunctions are not pre-attentive

Gestalt Grouping Principles

"All else being equal, elements that are related by X tend to be grouped perceptually into higher-order units."

— Stephen Palmer

- Proximity
- Similarity
- Connectedness
- Continuity
- Symmetry
- Closure
- Figure/Ground
- Common Fate

Change Blindness

- We don't always see everything that is there!
- Is the viewer able to perceive changes between two scenes?
  - If so, may be distracting
  - Can do things to minimize noticing changes

- Video: http://www.simonslab.com/videos.html

Summary of Design Criteria

- Choose expressive and effective encodings
  - Rule-based tests of expressiveness
  - Perceptual effectiveness rankings
    - Prioritizes encodings that are most easily/accurately interpreted
    - Principle of Importance Ordering: Encode more important information more effectively (Mackinlay)

Interaction

Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

Representation and Interaction

- Two main components of information visualization
- Very challenging to come up with innovative, new visual representations
- But can do interesting work with how user interacts with the view or views
  - Analysis is a process, often iterative with different interactions

"The effectiveness of information visualization hinges on two things: its ability to clearly and accurately represent information and our ability to interact with it to figure out what the information means."

S. Few, <Now you see it>

Taxonomy of Interactions

- Dix and Ellis (1998)
  - Highlighting and focus;
  - accessing extra info;
  - overview and context;
  - same representation, changing parameters;
  - Linking representations
- Keim (2002)
  - Projection
  - Filtering
  - Zooming
  - Distortion
  - Linking and brushing
- Few's Principles
  - Comparing
  - Sorting
  - Adding variables Filtering
  - Highlighting
  - Aggregating
  - Re-expressing
  - Re-visualizing
  - Zooming and panning
  - Re-scaling
  - Accessing details on demand
  - Annotating
  - Bookmarking

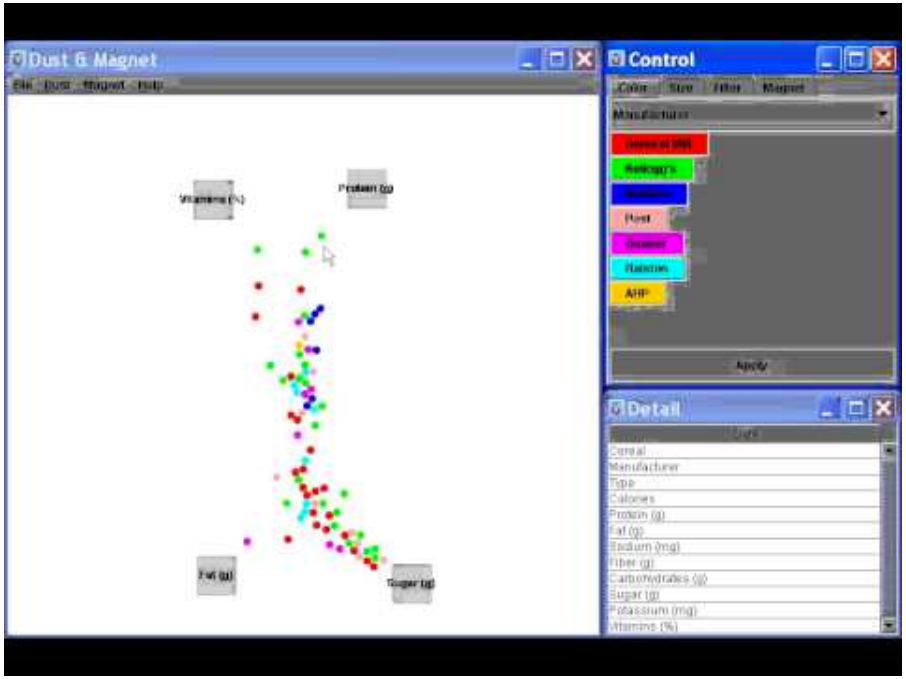
A Summary of Existing Taxonomy

- Survey
  - 59 papers
    - Papers introducing new interaction systems
    - Well-known papers in subareas of information visualization
  - 51 systems
    - Commercial Infovis Systems (SeeIT, Spotfire, TableLens, InfoZoom, etc.)
  - Collected 311 individual interaction techniques
- Affinity Diagram Method

Yi, Ji Soo, Youn ah Kang, and John Stasko. "Toward a deeper understanding of the role of interaction in information visualization." IEEE transactions on visualization and computer graphics 13.6 (2007): 1224-1231.

Taxonomy of Interactions based on User Intent

- 7 Categories
- Select
- Explore
- Reconfigure
- Encode
- Abstract/Elaborate
- Filter
- Connect



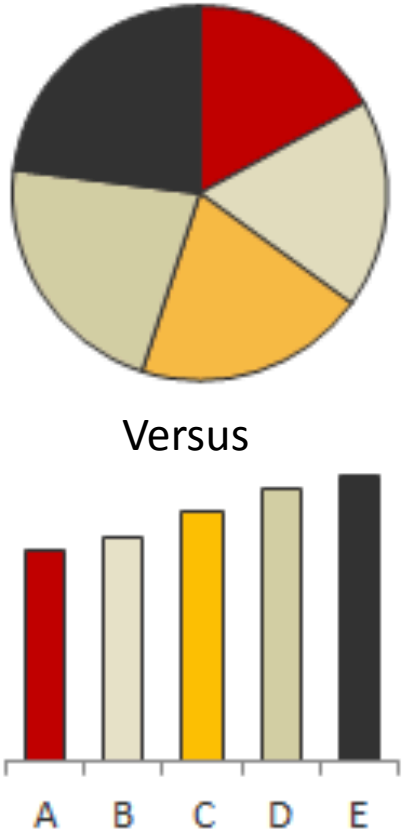
Interaction is Vital for Exploration

- Interaction facilitates a dialog between the user and the visualization system
- Multiple views amplify importance of interaction
- Interaction often helps when you just can't show everything you want

Evaluation

How do We Evaluate Visualizations?

- How do we evaluate visualizations?
  - Usability vs. Utility
- What evaluation techniques should we use?
- What do we measure?
  - What data do we gather?
  - What metrics do we use?

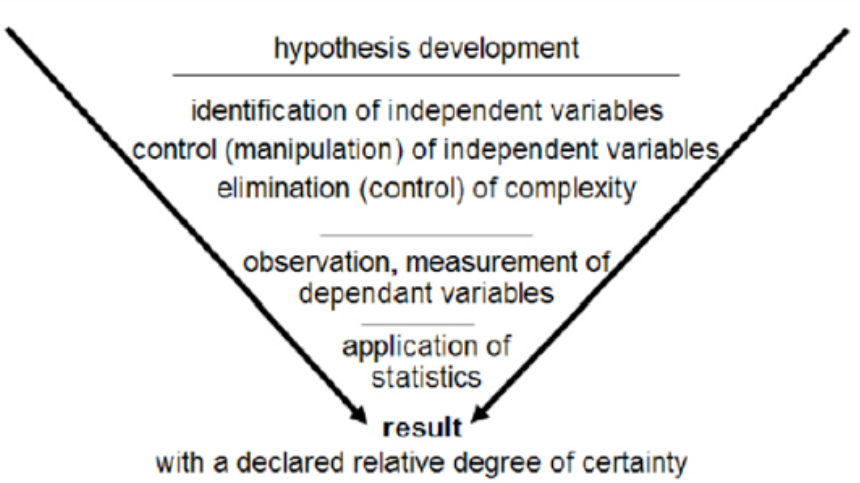


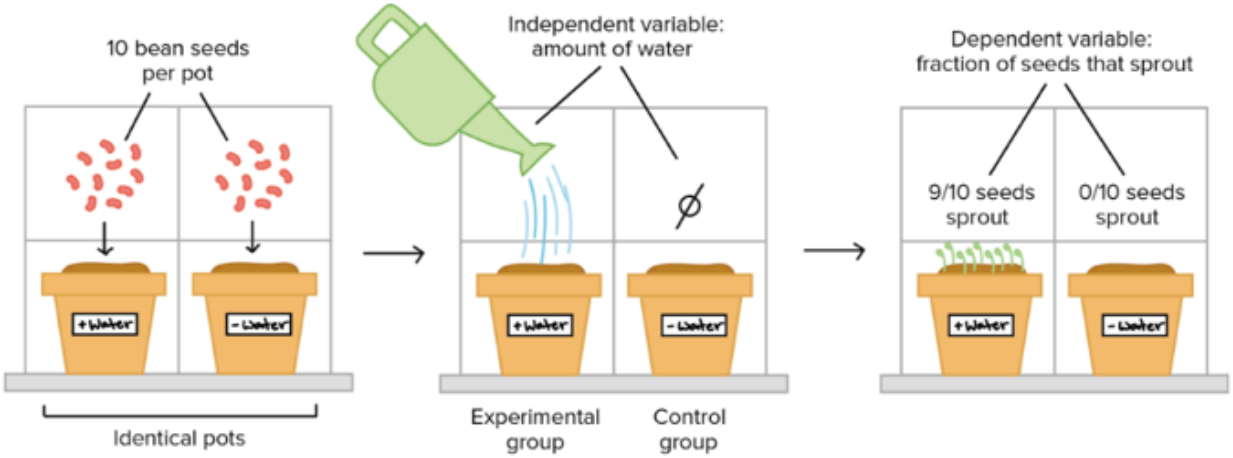
Evaluation Approaches

- Many different Forms
  - Qualitative, quantitative, objective, subjective, controlled experiments, interpretive observations, ...
- Two popular methodologies
  - Controlled experiments (Quantitative)
  - Subjective assessments (Qualitative)

Quantitative Methods: Controlled Experiments

- Good for measuring performance or comparing multiple techniques
- What do we measure?
  - Performance, time, errors, …





An Example: Controlled Experiment

- Run an experiment to evaluate three visualization strategies
  - Animation
  - Small multiples
  - Traces
- Especially interested in examining whether animated bubble charts are beneficial for analysis and presentation

Do you remember Hans Rosling's TED talk? (Lecture 2)

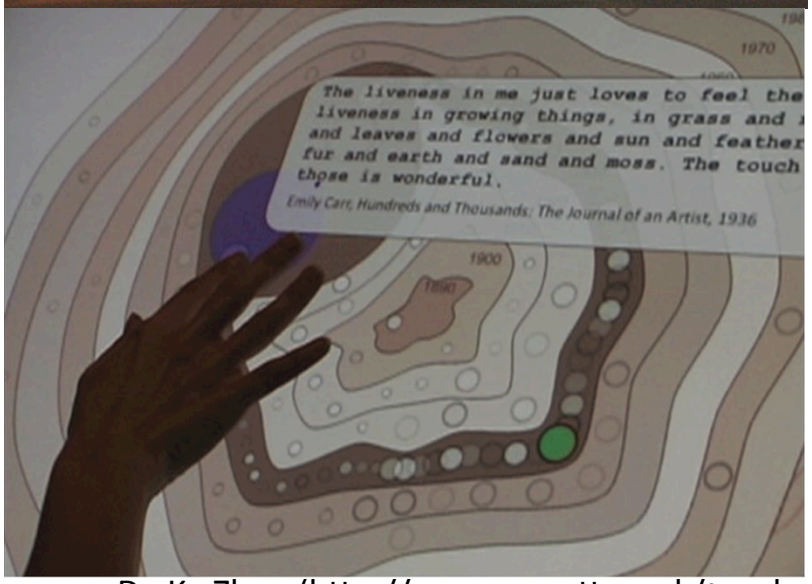
Qualitative Methods

- Types
  - Nested methods
    - Experimenter observation, think-aloud protocol, collecting participant opinions
  - Inspection evaluation methods
    - Heuristics to judge
- Observational context
  - In situ, laboratory, participatory
  - Contextual interviews is important

An Example: Subjective Assessments

- Evaluating a newly developed visualization system (EMDialog) at the museum (Emily Carr exhibit)

- Discourse visualization
  - Time
  - Context



Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

Methodology vs. Desirable Features

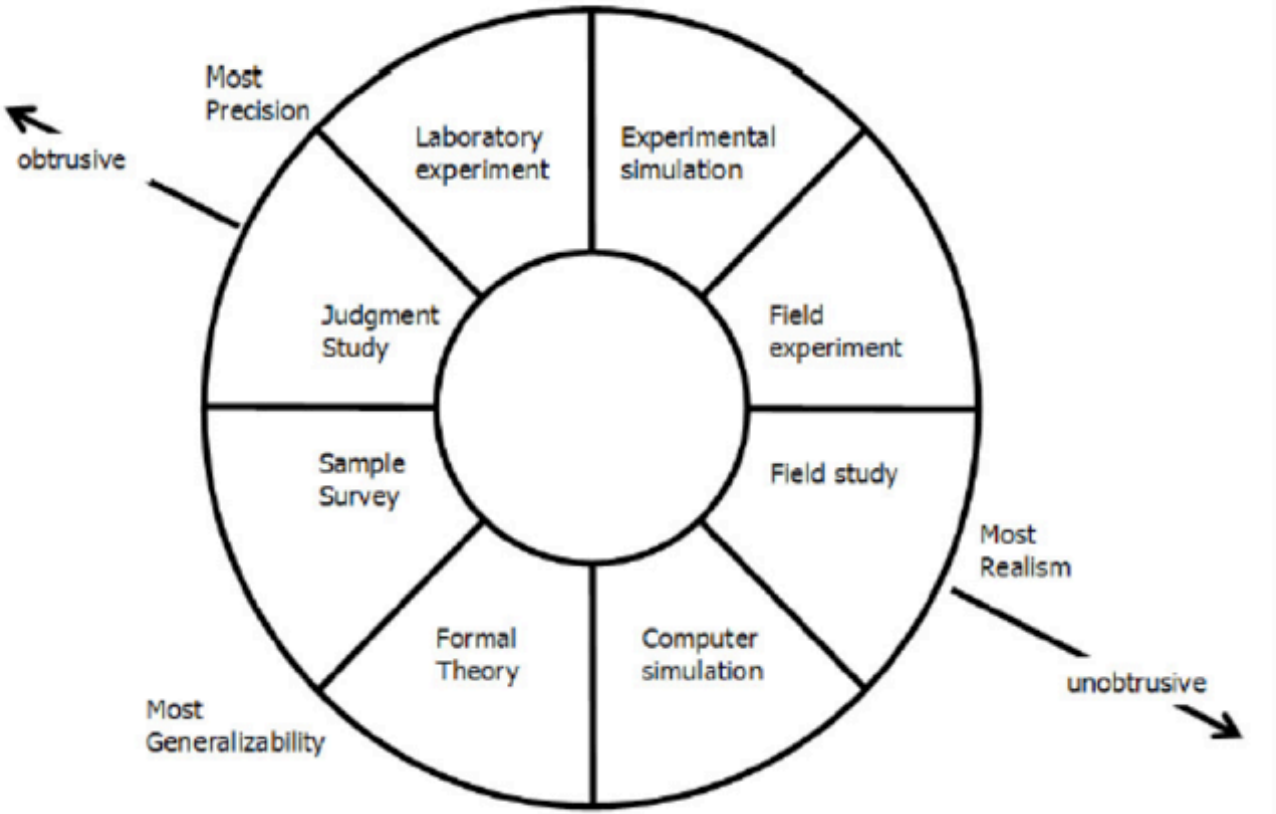
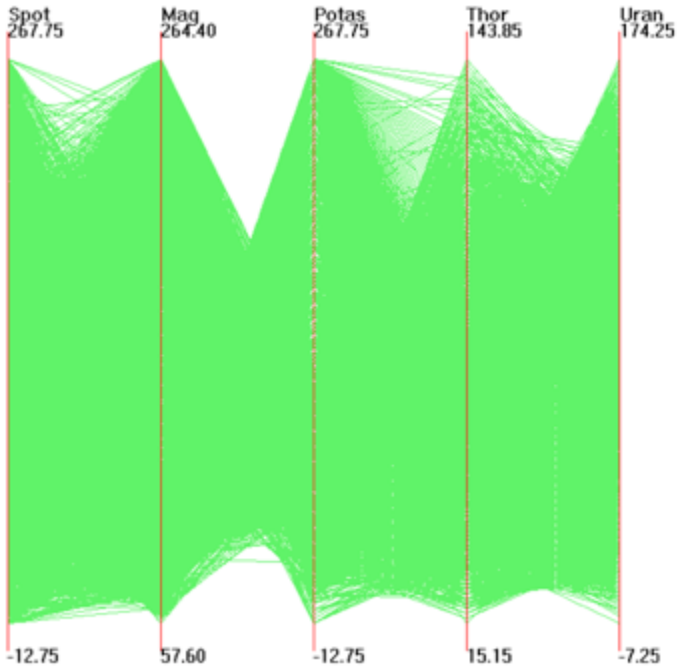


Fig. 1. Types of methodologies organized to show relationships to precision, generalizability and realism. (adapted, simplified from McGrath 1995)

Data Overload

Data Overload

- Most of the techniques we've examined work for a modest number of data cases or variables
- What happens when you have lots and lots of data cases and/or variables?



Out5d dataset(5 dimensions, 16384 items)

Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

General Solution

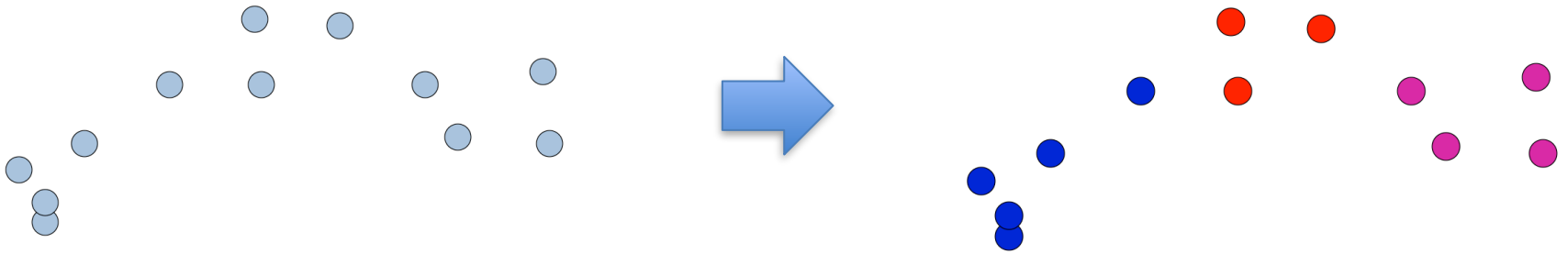
- Data that is similar in most dimensions ought to be drawn together
  - Cluster at high dimensions
- Need to project the data down into the plane and give it some ultra-simplified representation
- Or perhaps only look at certain aspects of the data at any one time

Clustering and Dimensionality Reduction

- There exist many techniques for clustering high-dimensional data with respect to all those dimensions (too many data points)
  - Affinity propagation
  - k-means
  - Expectation maximization
  - Hierarchical clustering

- There exist many techniques for projecting n-dimensions down to 2-D (dimensionality reduction, too many variables)
  - Multi-dimensional scaling (MDS)
  - Principal component analysis (PCA)
  - Linear discriminant analysis
  - Factor analysis

K-means Clustering



- The most well-known and popular clustering algorithm

K-means Algorithm

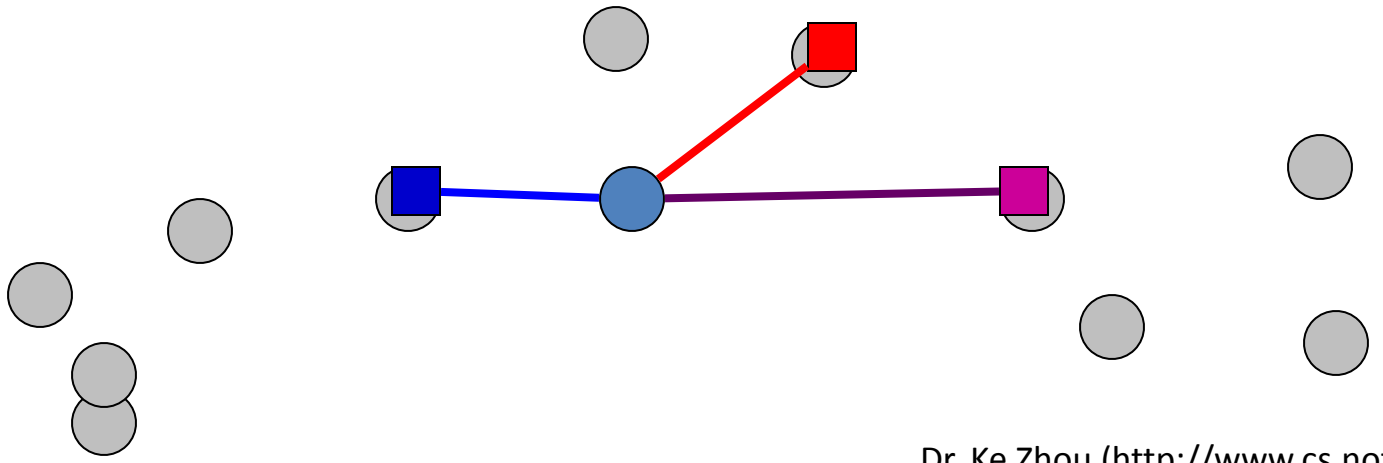
Iterate:

- Start with some initial cluster centers
- Assign/cluster each example to closest center

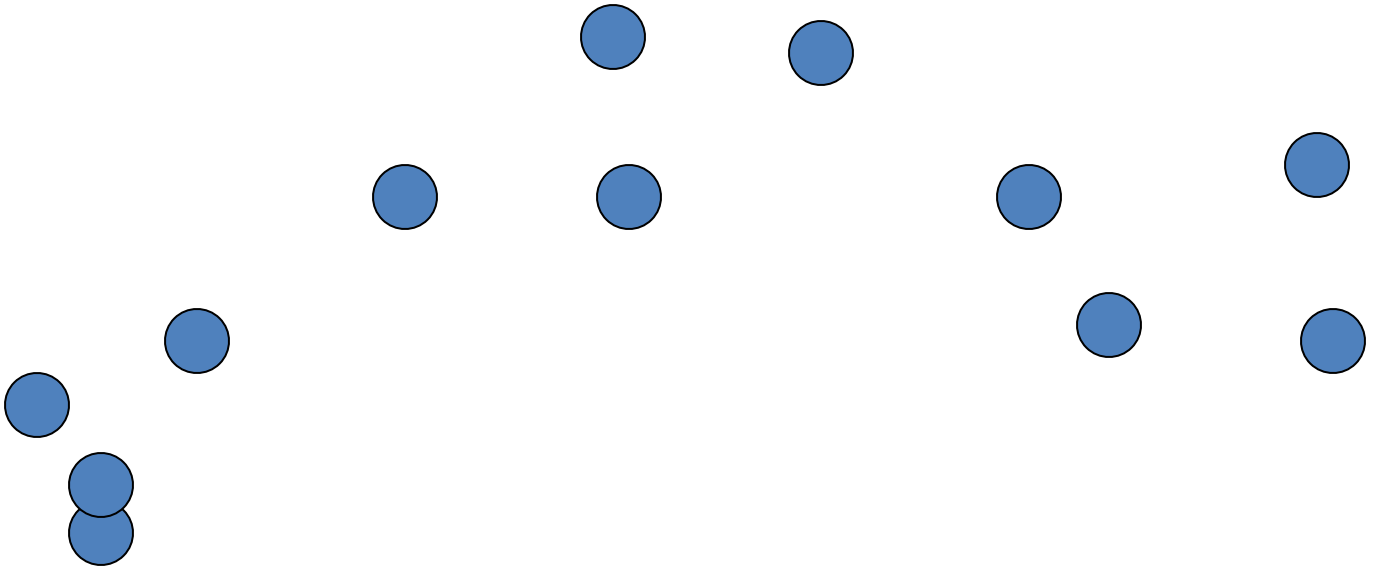
  iterate over each point:

  - get distance to each cluster center
  - assign to closest center (hard cluster)

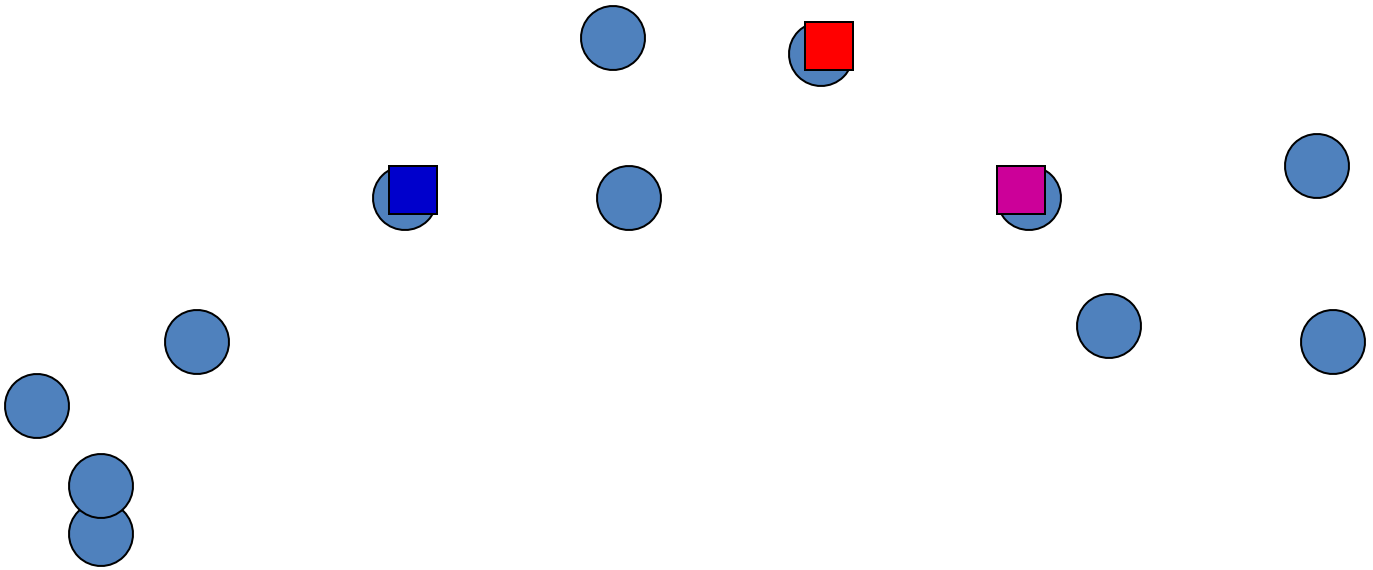
- Recalculate centers as the mean of the points in a cluster



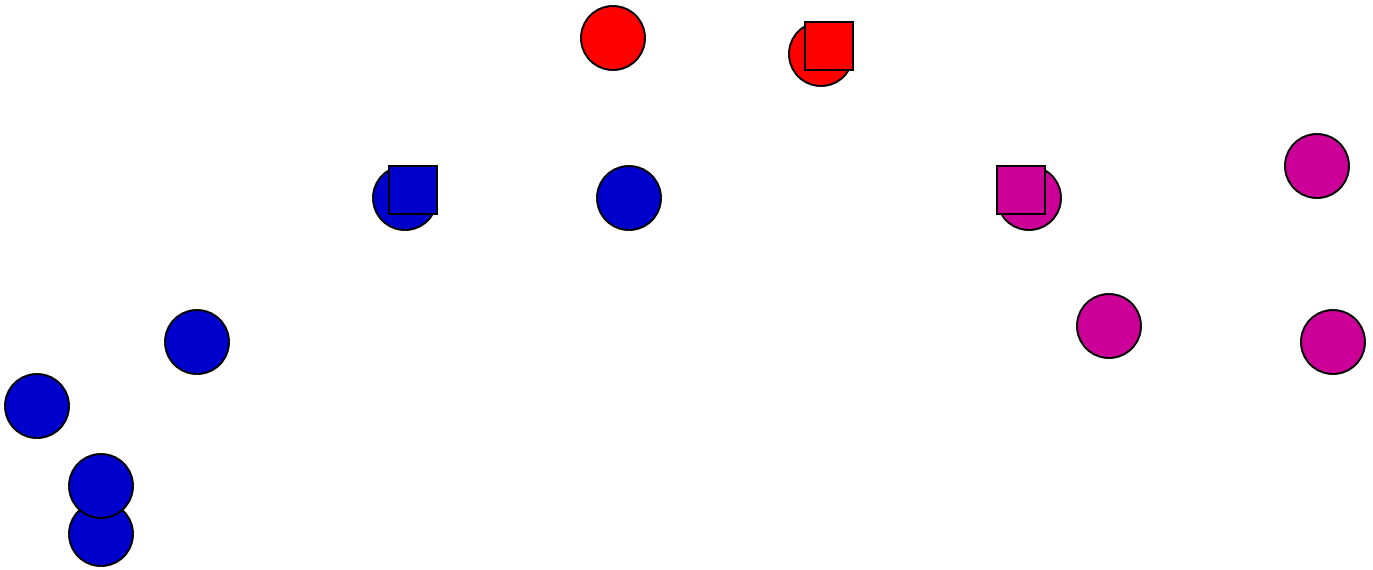
K-means: an example



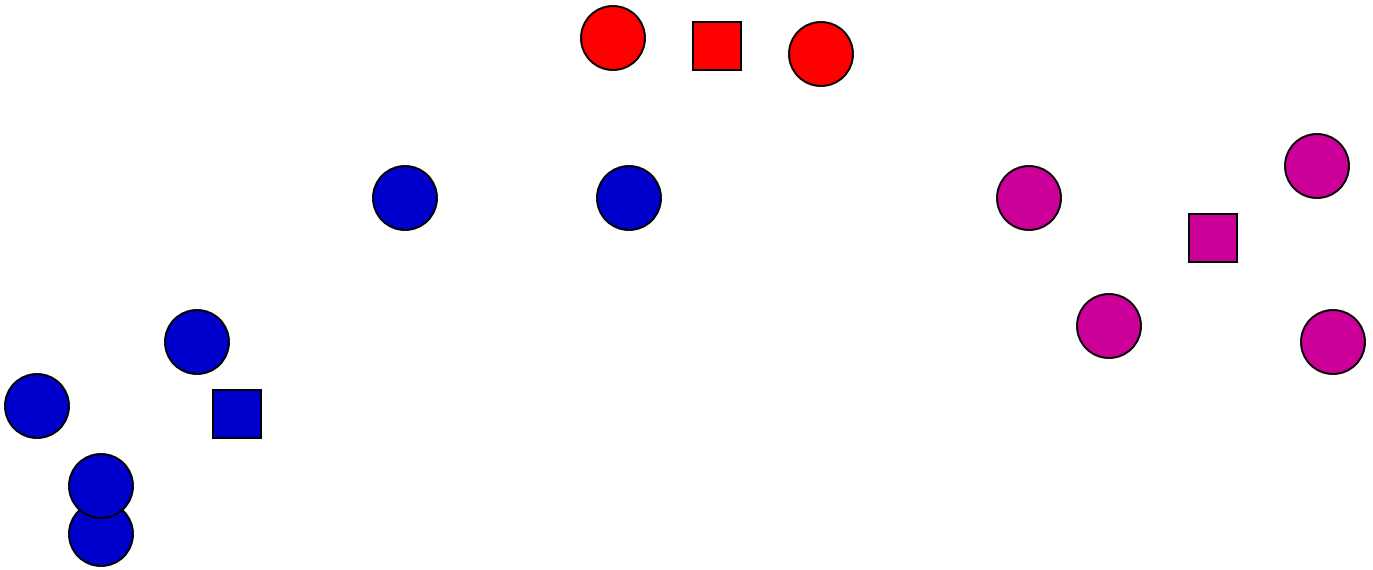
K-means: Initialize centers randomly



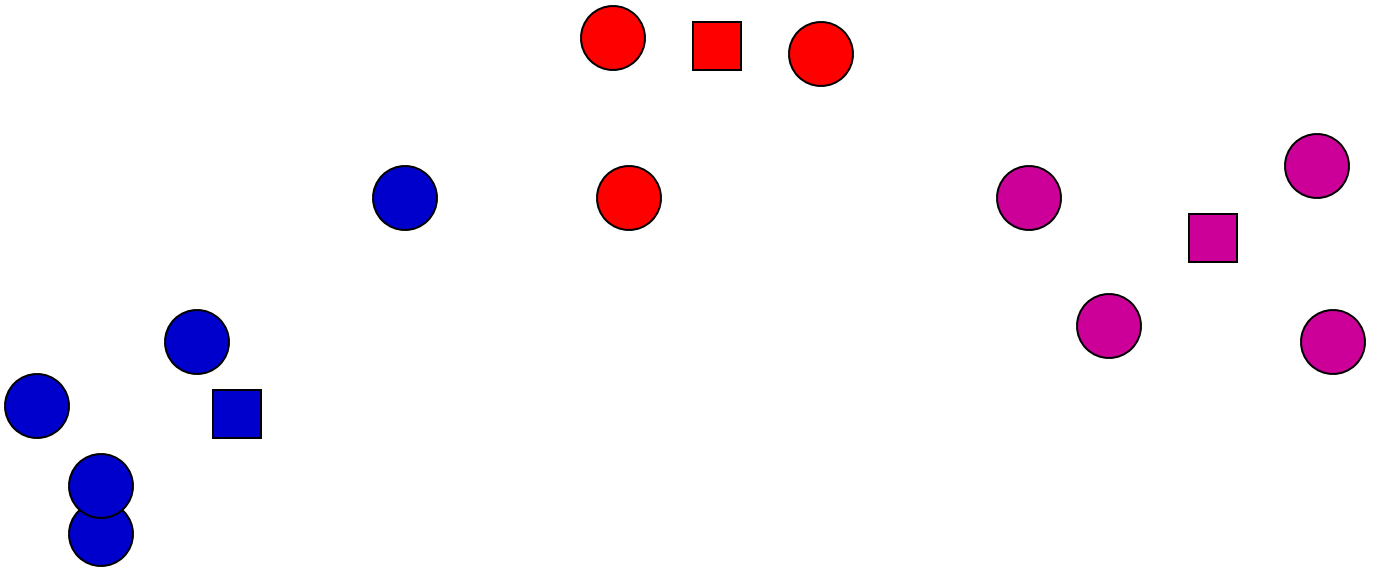
K-means: assign points to nearest center



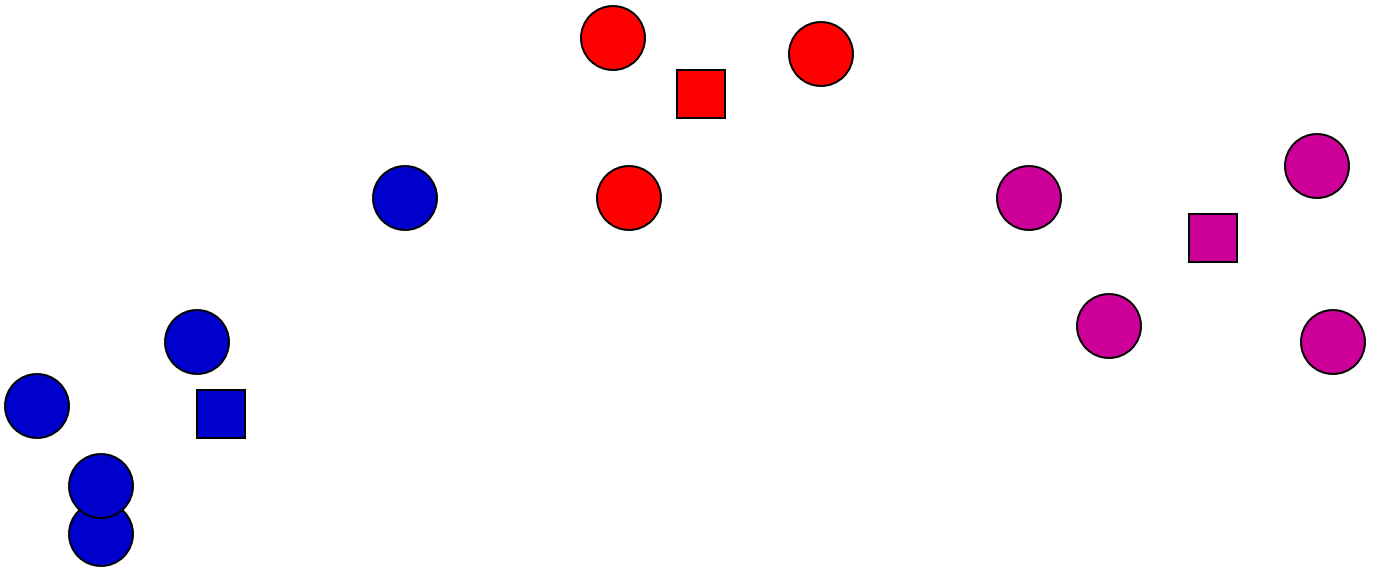
K-means: readjust centers



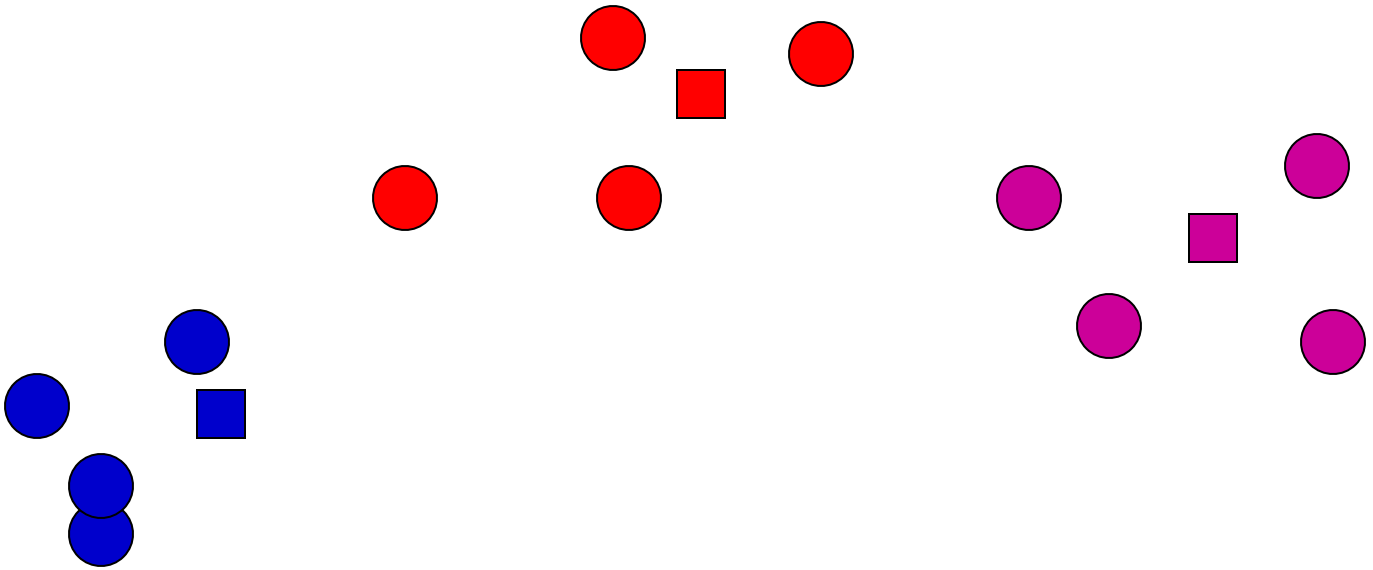
K-means: assign points to nearest center



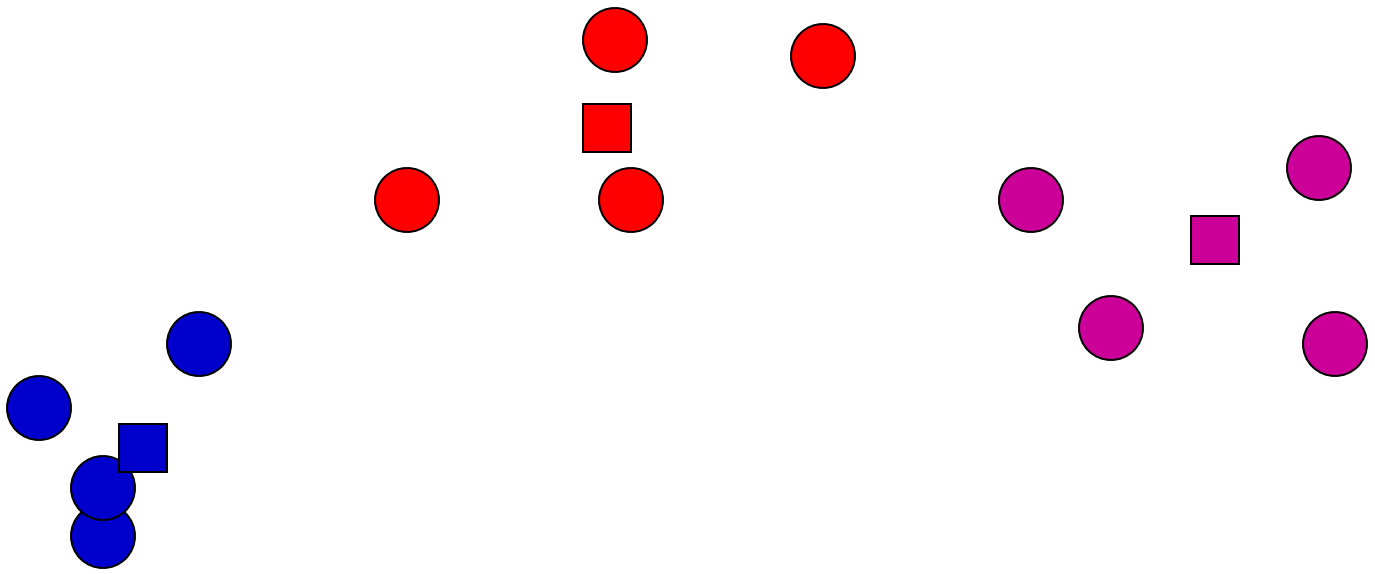
K-means: readjust centers



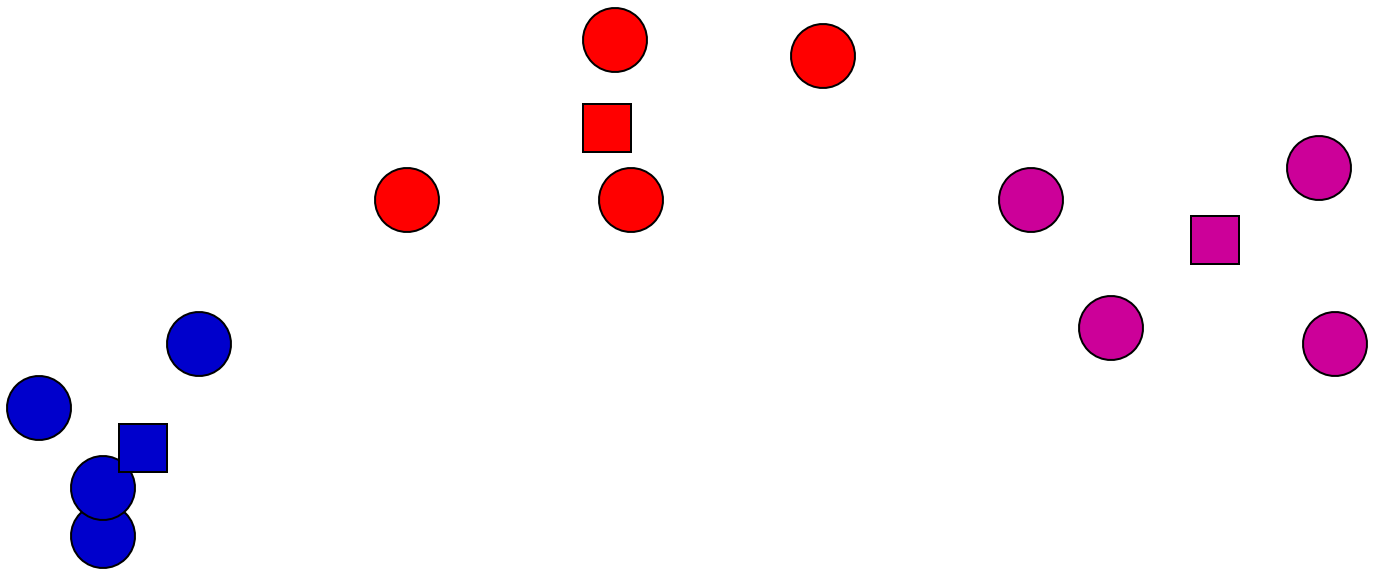
K-means: assign points to nearest center



K-means: readjust centers



K-means: assign points to nearest center



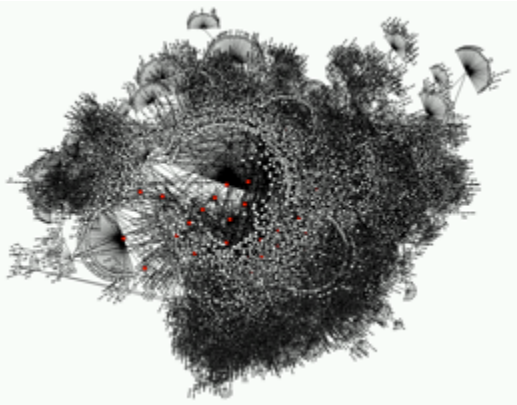
No changes: Done

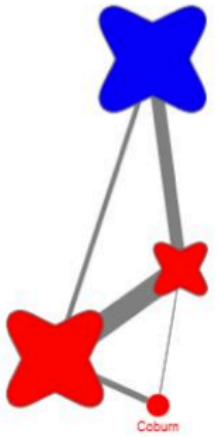
Other Reduction Techniques

- Other techniques exist to manage scale
  - Sampling: only including every so many data cases or variables
  - Aggregation: combining many data cases or variables
- Interaction
  - Employ user interaction rather than special renderings to help manage scale

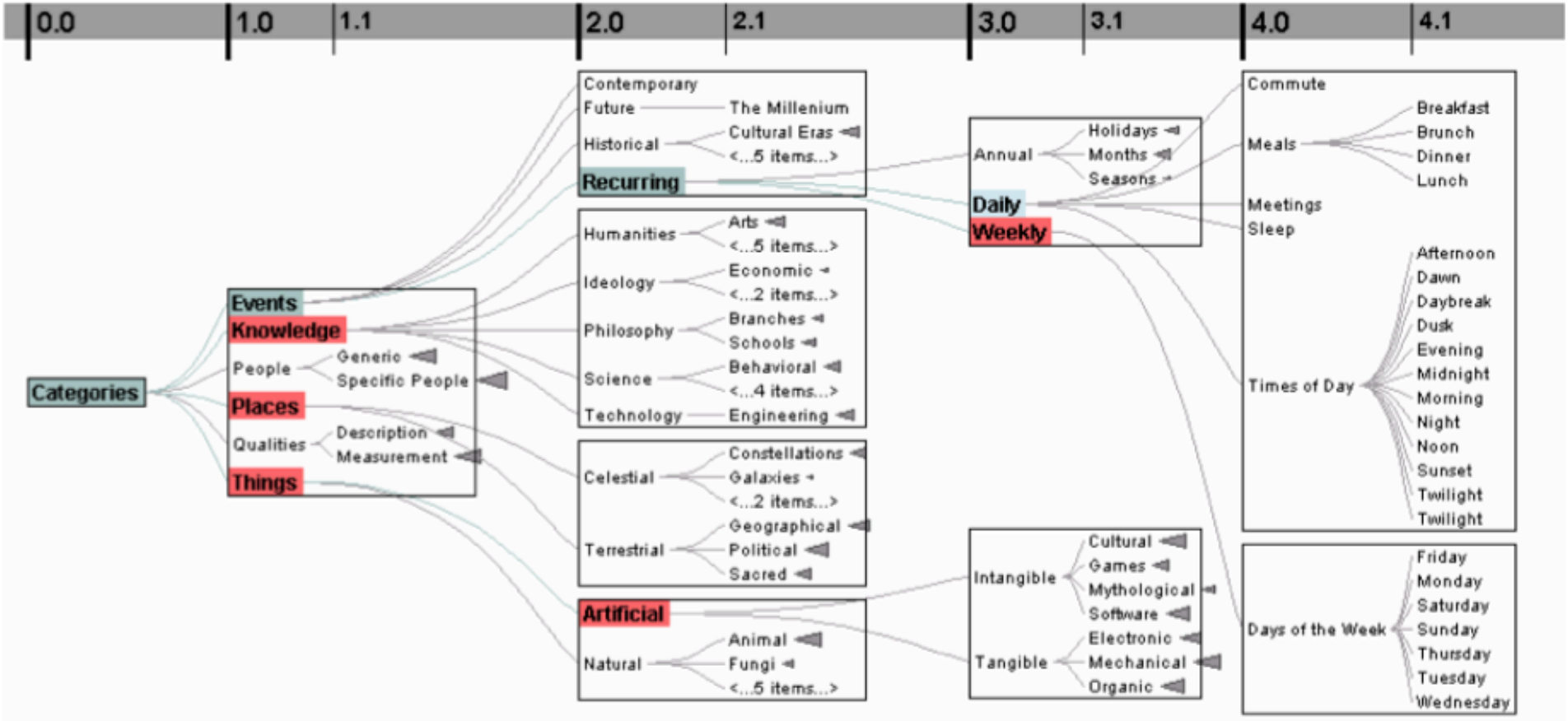
Application: Scalability Issue of Graphs

- Need to cope with messiness
- Solutions
  - Extracting network motifs
  - Taking advantage of node attributes
  - Degree-of-Interest graphs
  - Use the alternative representation: matrix





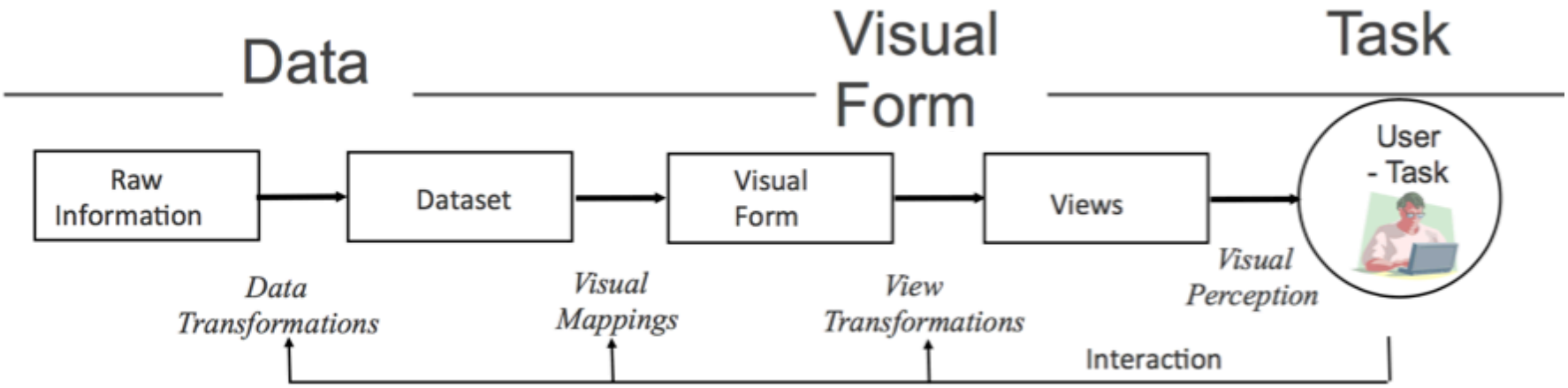
Interactive: Degree-of-interest Trees/Graphs



- Cull “un-interesting” nodes on a per block basis until all blocks on a level fit within bounds.
- Attempt to center child blocks beneath parents.

Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

Information Visualization



Next Lecture

- Topic:
  - Review