G53FIV: Fundamentals of Information Visualization

Lecture 4: Design and Graphs

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

https://moodle.nottingham.ac.uk/course/view.php?id=96914

Last Lecture

Data and Image

Overview

- Design Criteria
- Graphs
  - for uni, bi and tri-variate data

What design criteria should we follow?

Choosing Visual Encodings

- Assume k visual encodings and n data attributes. We would like to pick the “best” encoding among a combinatorial set of possibilities of size $(n+1)^k$
- Principle of Consistency
  - The properties of the image (visual variables) should match the properties of the data.
- Principle of Importance Ordering
  - Encode the most important information in the most effective way.

The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

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

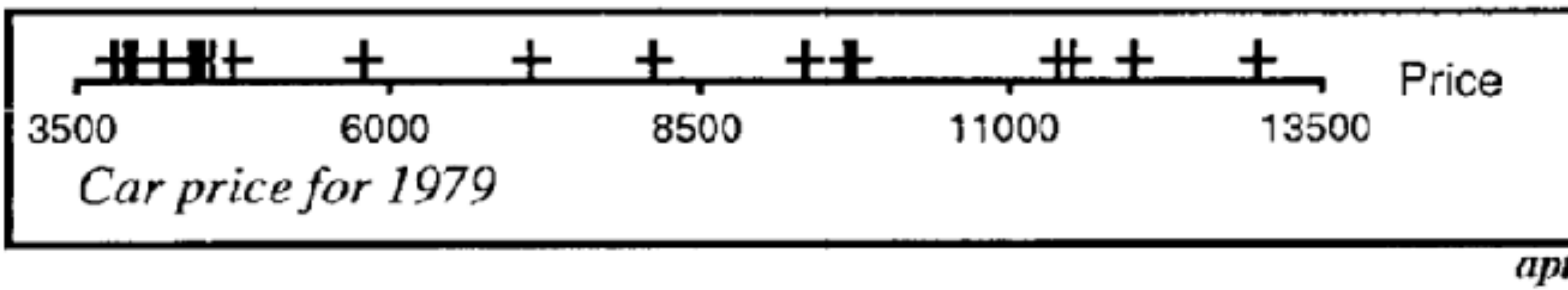
- **Effectiveness**
  - A visualization is more effective than another visualization if the information conveyed by one visualization is more readily perceived than the information in the other visualization.

Use proper encoding

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

Expressiveness

- Unable to express all facts in a layout (fails first criterion)

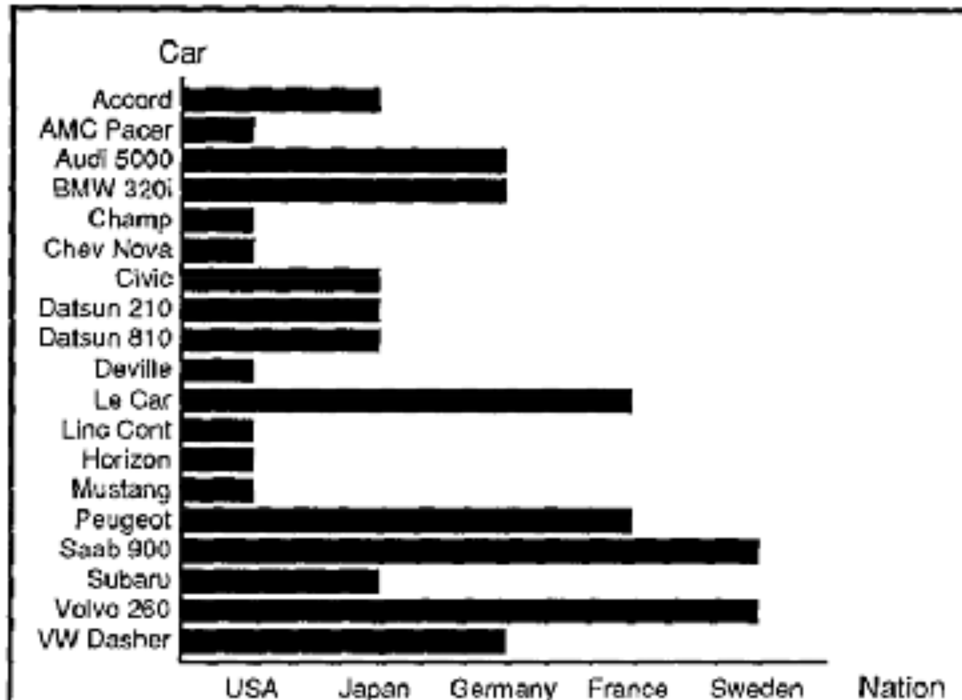


Price :	*Cars*	→ [3500, 13000]
Mileage :	*Cars*	→ [10, 40]
Weight :	*Cars*	→ [1500, 5000]
Repair :	*Cars*	→ ⟨Great, Good, OK, Bad, Terrible⟩
Nation :	*Cars*	→ {USA, Germany, France, ⋯}

Cars = {Accord, AMC Pacer, Audi 5000, BMW 320i, ⋯}

Expressiveness

- Expresses information not inherent in the dataset (fails second criterion)

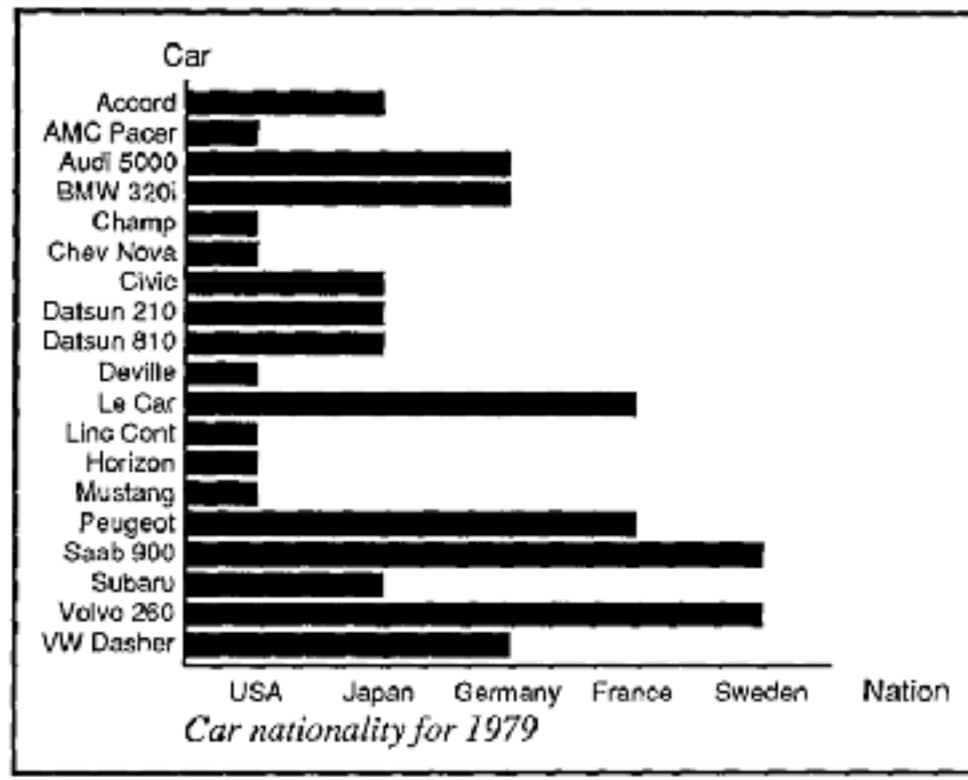


Car nationality for 1979

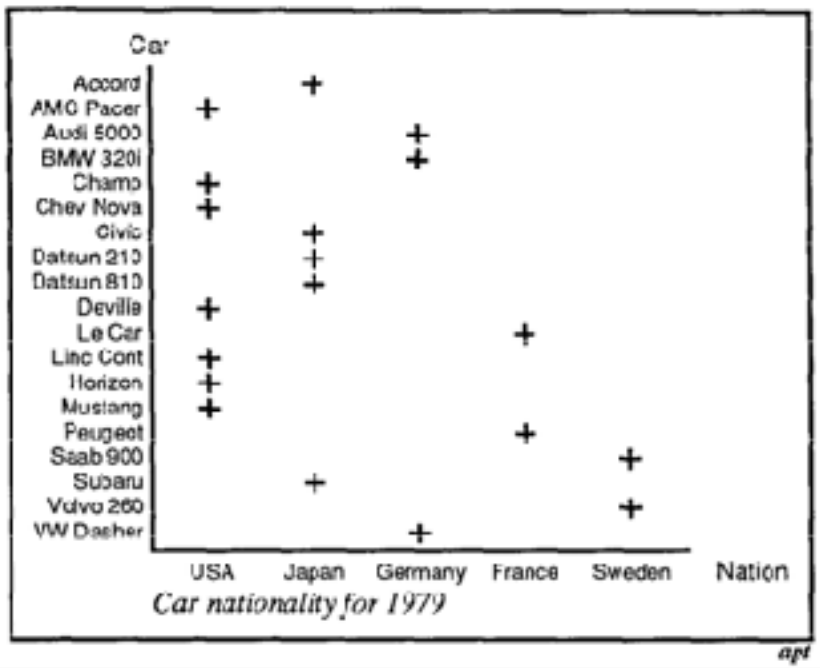
The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

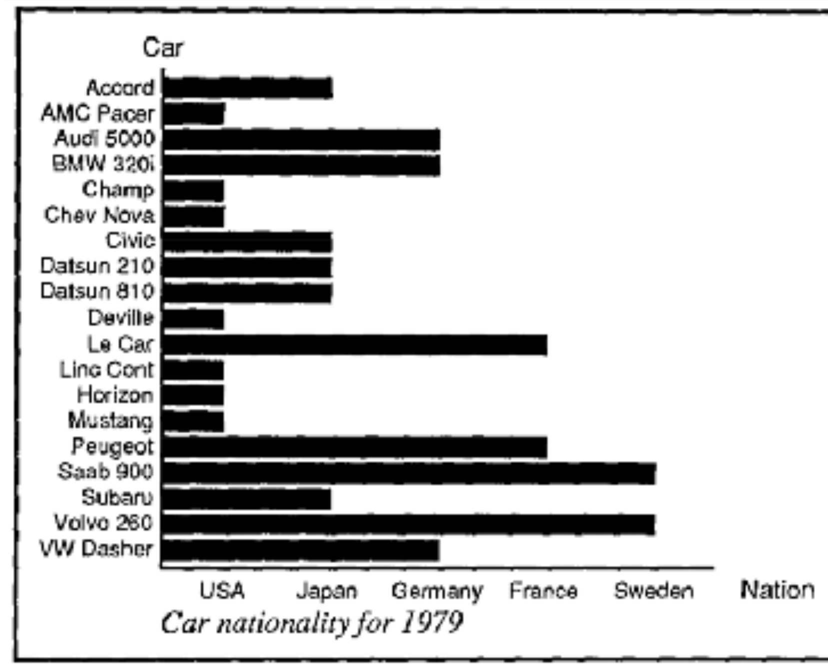
Expressiveness

- Expresses information not inherent in the dataset (fails second criterion)
- A length is interpreted as a quantitative value.



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.

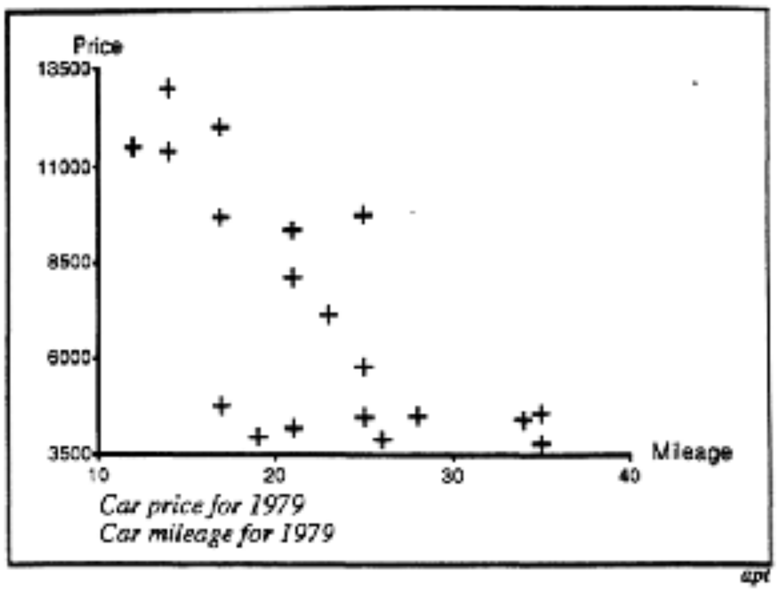
Tell the truth

- **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.

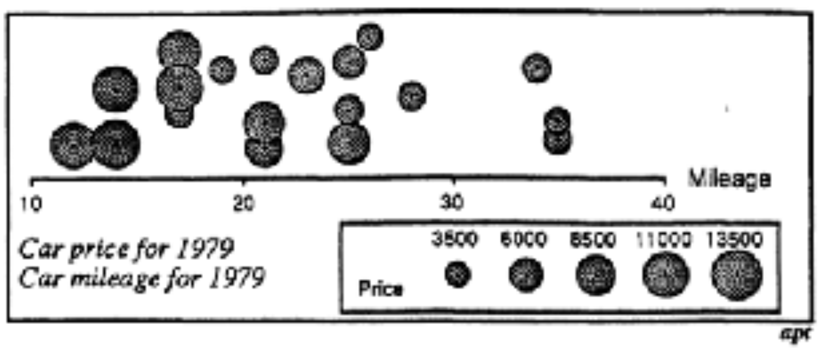
Use proper encoding

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

Effectiveness

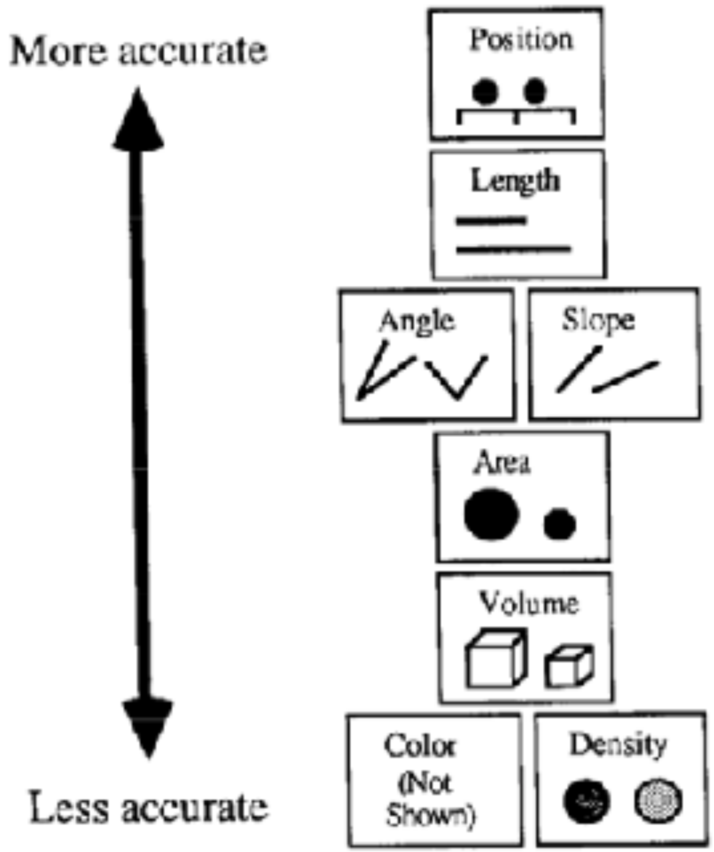


Vs.



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

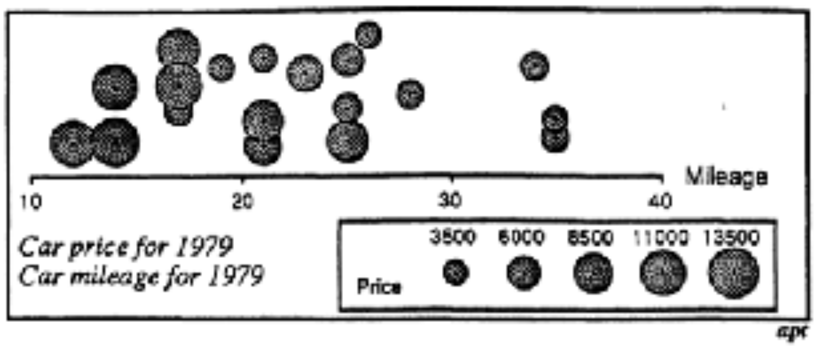
Effectiveness: Accuracy Ranking for Quantitative Information

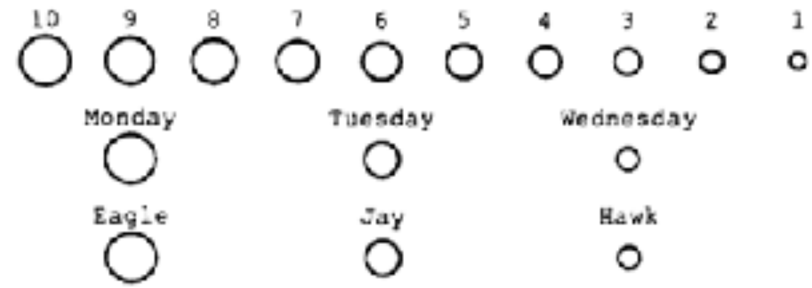


Mackinlay, Automating the design of graphical presentations of relational information, 1986.

Effectiveness: Accuracy Ranking for Nominal/Ordinal Information?

Area Encoding





Quantitative

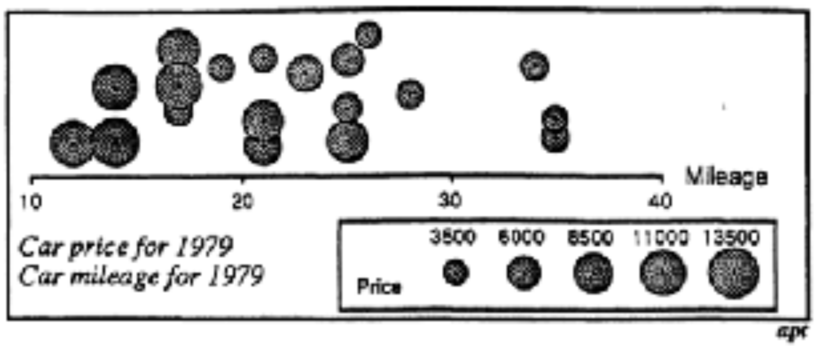
Nominal

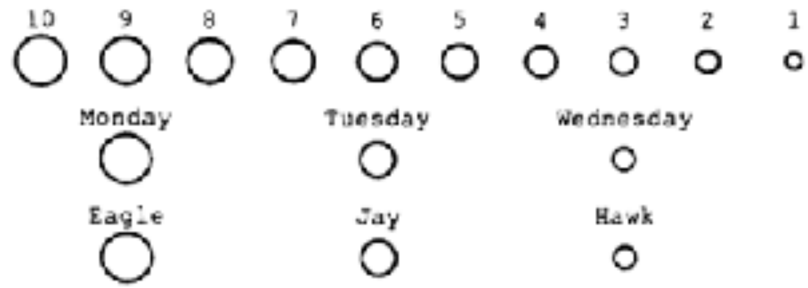
We can use, but not so accurate.

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

Effectiveness: Accuracy Ranking for Nominal/Ordinal Information?

Area Encoding





Quantitative

We can use, but not so accurate.

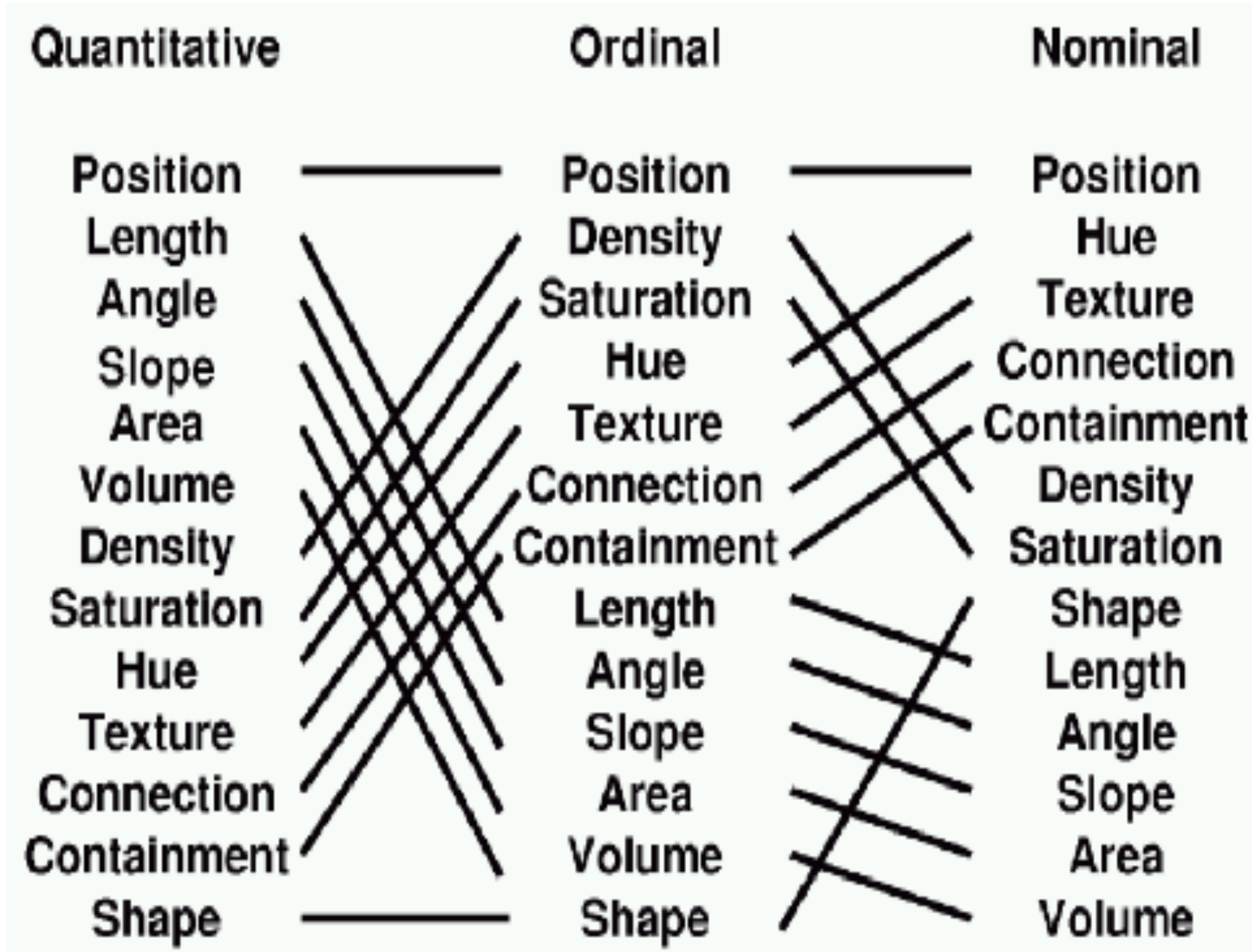
Nominal

- Problematic if there are too many categories;
- Can be expected to encode ordinal 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.

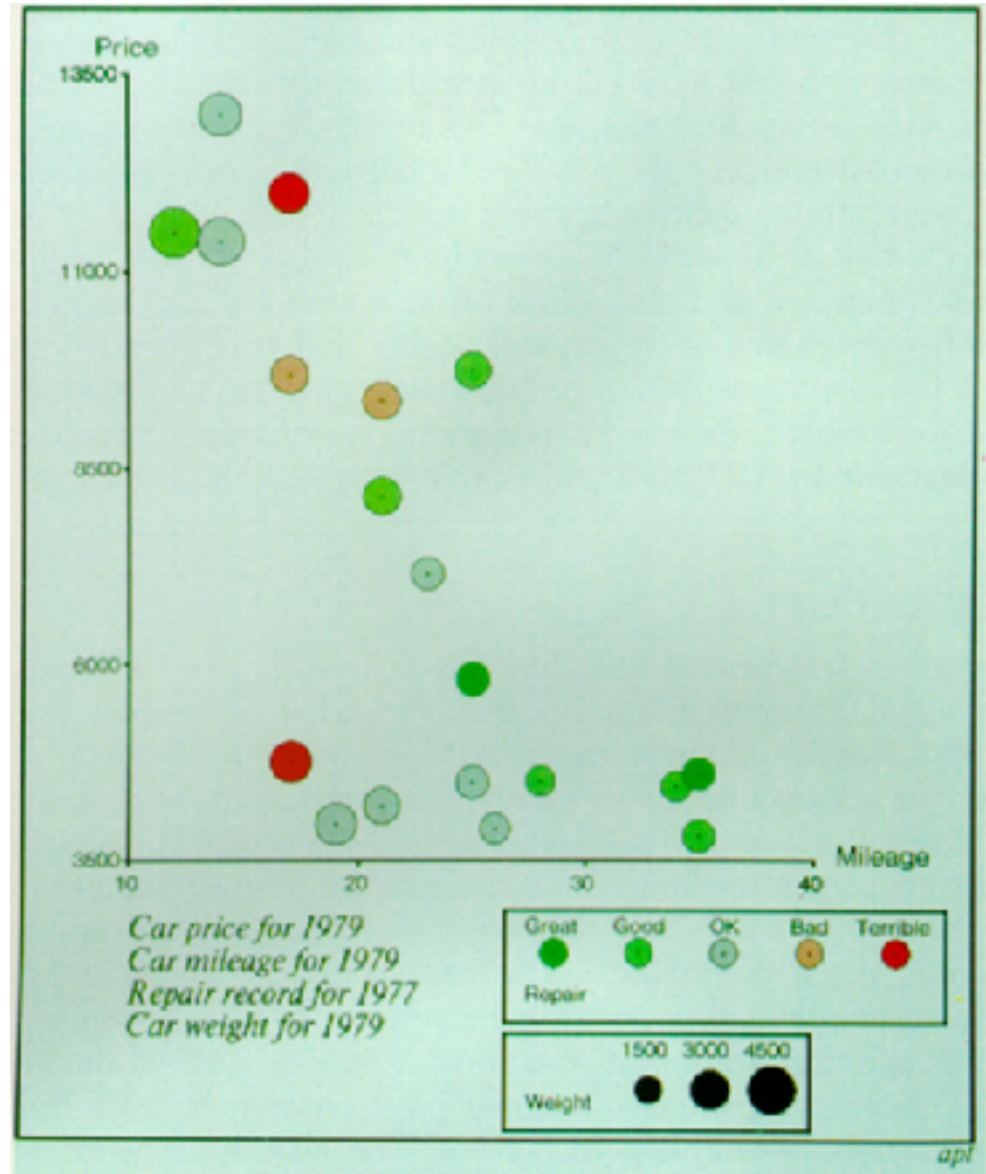
Mackinlay’s Design Algorithm

- APT - “A Presentation Tool”, 1986
- User formally specifies data model and type
  - Input: ordered list of data variables to show
- APT searches over design space
  - Test expressiveness of each visual encoding Generate encodings that pass test
  - Rank by perceptual effectiveness criteria
- Output the “most effective” visualization

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

APT

- Automatically generate chart for car data
- Input variables:
  - Price
  - Mileage
  - Repair
  - Weight



zkz/)

Limitations of APT?

- Does not cover many visualization techniques
  - Networks, hierarchies, maps, diagrams
  - Also: 3D structure, animation, illustration, ...
- Does not consider interaction
- Does not consider semantics / conventions
- Assumes single visualization as output

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)
- Question: how do we establish effectiveness criteria?
  - Subject of the visual perception lecture...

Graphs

How Many Variables?

- Data sets of dimensions 1, 2, 3 are common
- Number of variables per class
  - 1 - Univariate data
  - 2 - Bivariate data
  - 3 - Trivariate data
  - >3 - Hypervariate data

Graphs

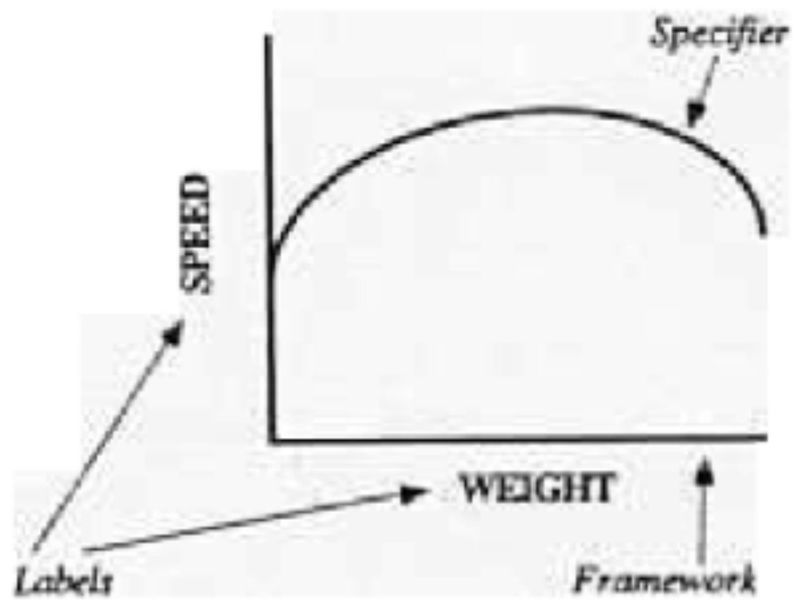
- Data Dimensions
  - 1 - Univariate data
  - 2 - Bivariate data
  - 3 - Trivariate data
  - >3 - Hypervariate data

We mainly focus on uni, bi and tri-variate data for the rest of the lecture.

- Data Types
  - Nominal, Ordinal, Quantitative
- Visualization Representations
  - Points, Lines, Bars, Boxes

Components of Graphs

- Framework
  - Measurement types, scale
- Content (Specifier)
  - Marks, lines, points
- Labels
  - Title, axes, ticks

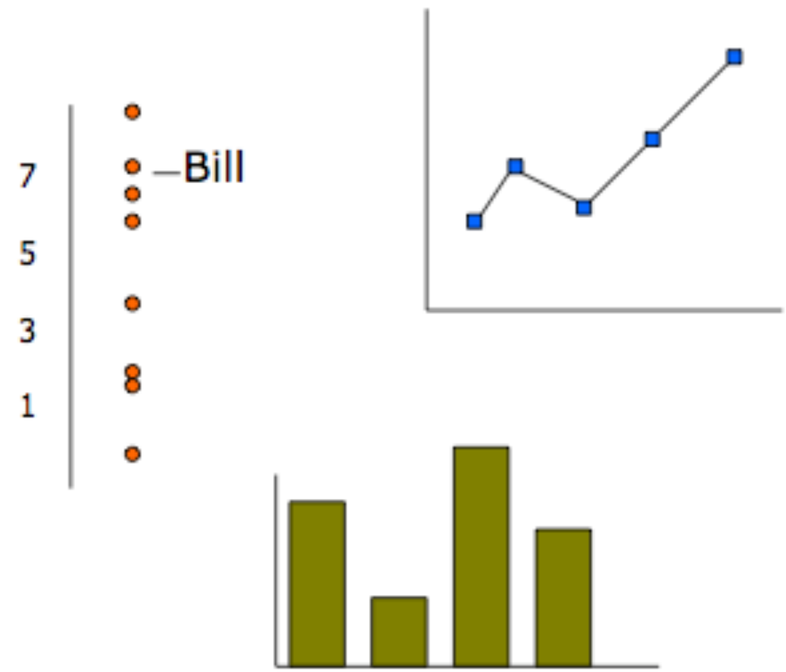


Points, Lines, Bars, Boxes

- Points
  - Useful in scatterplots for 2-values
  - Can replace bars when scale doesn’t start at 0
- Lines
  - Connect values in a series
  - Show changes, trends, patterns
  - Not for a set of nominal or ordinal values
- Bars
  - Emphasizes individual values
  - Good for comparing individual values
- Boxes
  - Shows a distribution of values

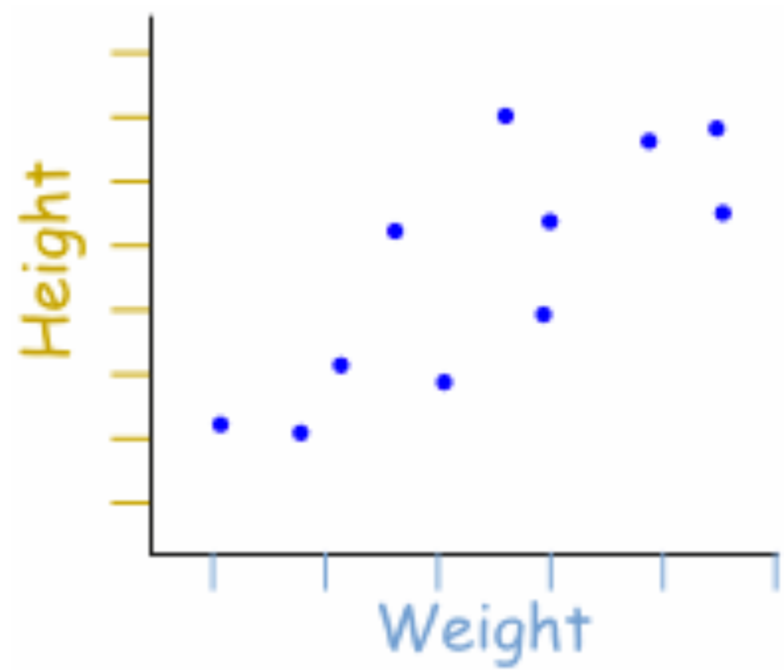
Univariate Data

- 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)



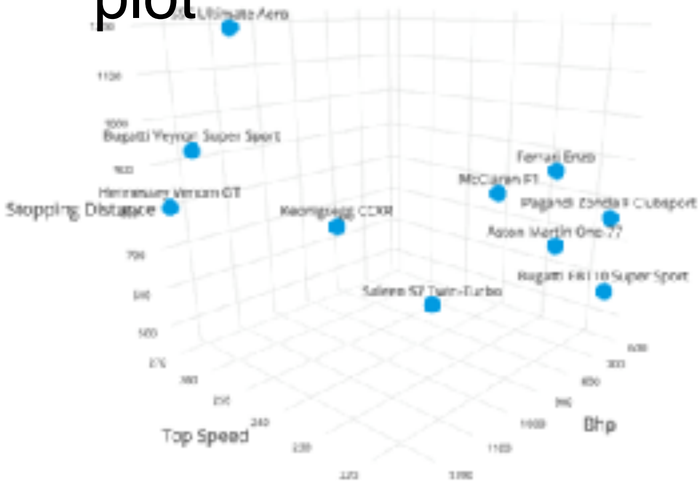
Bivariate Data

- 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?

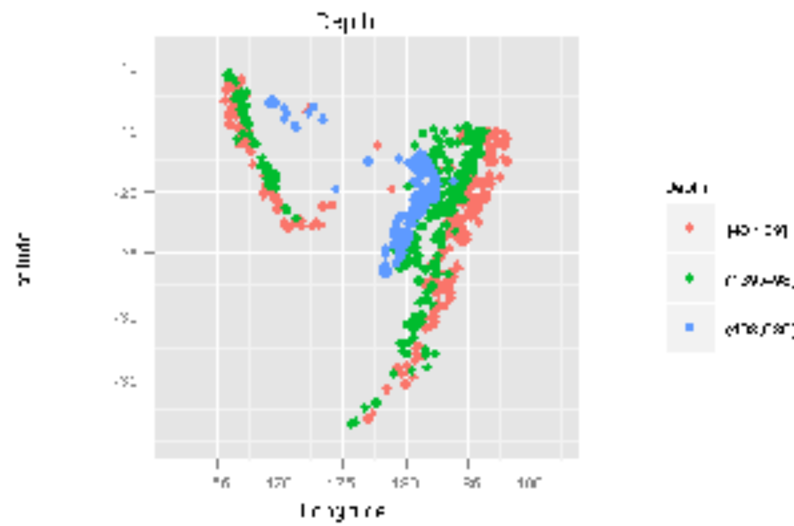


Trivariate Data

3D scatter plot

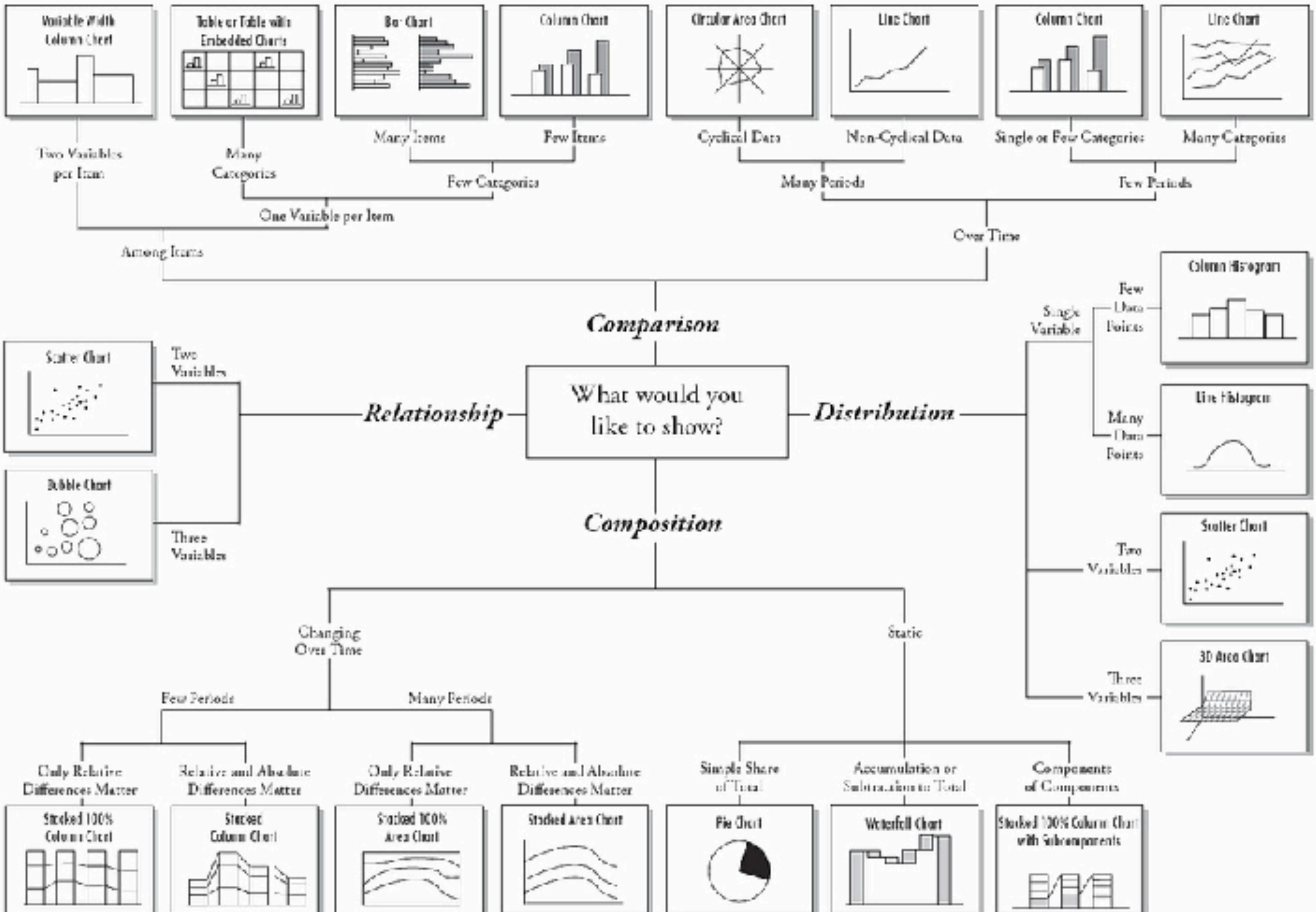


2D + mark property



Represent each variable in its own explicit way

Chart Suggestions—A Thought-Starter



Data Visualization with ggplot2 : : CHEAT SHEET

Basics

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

To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.

Complete the template below to build a graph.

ggplot (data = <DATA>) + <GEOM_FUNCTION> (mapping = aes(<MAPPINGS>), stat = <STAT> , position = <POSITION>) + <COORDINATE_FUNCTION> + <FACET_FUNCTION> + <SCALE_FUNCTION> + <THEME_FUNCTION>

required

Not required, sensible defaults supplied

ggplot(data = mpg, **aes**(x = cty, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

aesthetic mappings · data · geom

qplot(x = cty, y = hwy, data = mpg, geom = "point") Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

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

GRAPHICAL PRIMITIVES

a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + **geom_blank()**
(Useful for expanding limits)

b + **geom_curve(aes**(yend = lat + 1, xend=long+1,curvature=z)) - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

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

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

b + **geom_rect**(aes(xmin = long, ymin=lat, xmax= long + 1, ymax = lat + 1)) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

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

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

b + **geom_abline**(aes(intercept=0, slope=1))
b + **geom_hline**(aes(yintercept = lat))
b + **geom_vline**(aes(xintercept = long))

b + **geom_segment**(aes(yend=lat+1, xend=long+1))
b + **geom_spoke**(aes(angle = 1:1155, radius = 1))

ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + **geom_area(stat = "bin")**
x, y, alpha, color, fill, linetype, size

c + **geom_density**(kernel = "gaussian")
x, y, alpha, color, fill, group, linetype, size, weight

c + **geom_dotplot()**
x, y, alpha, color, fill

c + **geom_freqpoly()** x, y, alpha, color, group, linetype, size

c + **geom_histogram**(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight

c2 + **geom_qq**(aes(sample = hwy)) x, y, alpha, color, fill, linetype, size, weight

discrete

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

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

TWO VARIABLES

continuous x , continuous y

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

e + **geom_label**(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE) x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + **geom_jitter**(height = 2, width = 2)
x, y, alpha, color, fill, shape, size

e + **geom_point()**, x, y, alpha, color, fill, shape, size, stroke

e + **geom_quantile()**, x, y, alpha, color, group, linetype, size, weight

e + **geom_rug**(sides = "bl"), x, y, alpha, color, linetype, size

e + **geom_smooth**(method = lm), x, y, alpha, color, fill, group, linetype, size, weight

e + **geom_text**(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE), x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

discrete x , continuous y

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

f + **geom_col()**, x, y, alpha, color, fill, group, linetype, size

f + **geom_boxplot()**, x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

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

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

discrete x , discrete y

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

g + **geom_count()**, x, y, alpha, color, fill, shape, size, stroke

continuous bivariate distribution

h <- ggplot(diamonds, aes(carat, price))

h + **geom_bin2d**(binwidth = c(0.25, 500))
x, y, alpha, color, fill, linetype, size, weight

h + **geom_density2d()**
x, y, alpha, colour, group, linetype, size

h + **geom_hex()**
x, y, alpha, colour, fill, size

continuous function

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

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

i + **geom_line()**
x, y, alpha, color, group, linetype, size

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

visualizing error

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

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

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

j + **geom_linerange()**
x, ymin, ymax, alpha, color, group, linetype, size

j + **geom_pointrange()**
x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

visualizing error

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

k + **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)); l <- ggplot(seals, aes(long, lat))

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

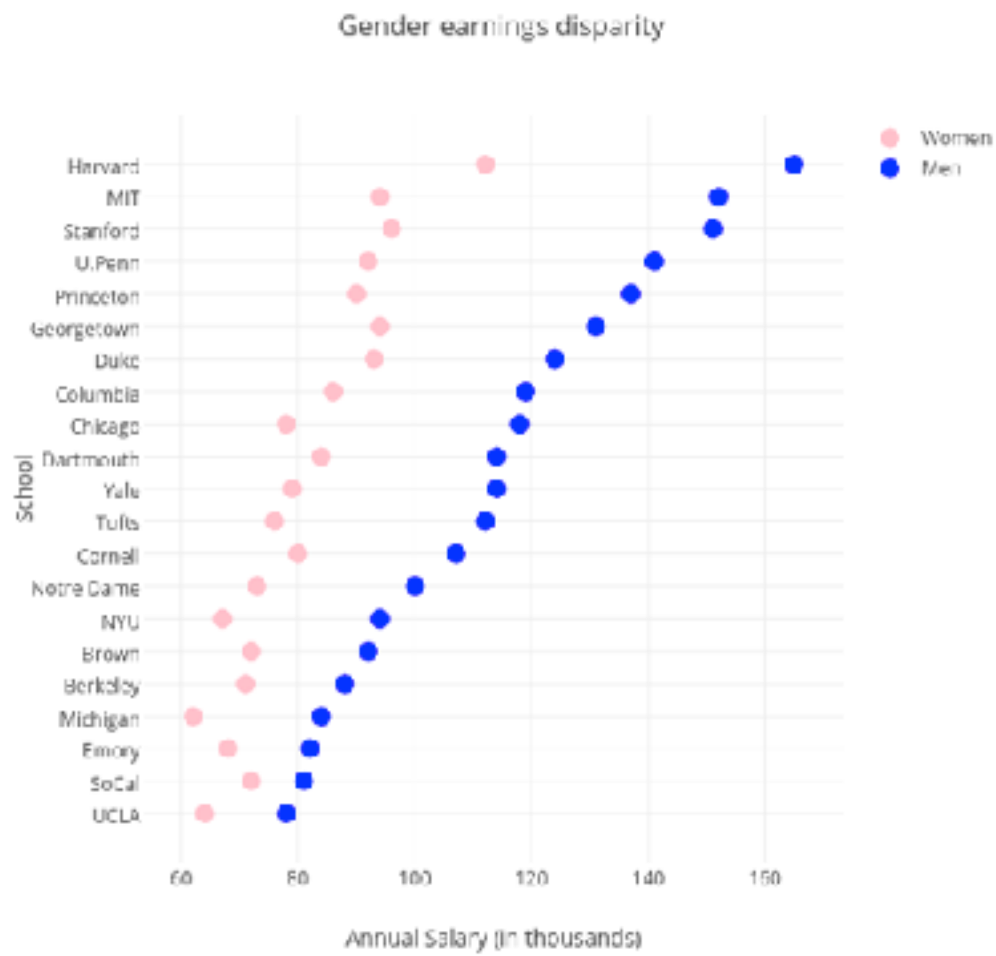
l + **geom_raster**(aes(fill = z), hjust=0.5, vjust=0.5, interpolate=FALSE)
x, y, alpha, fill

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

RStudio

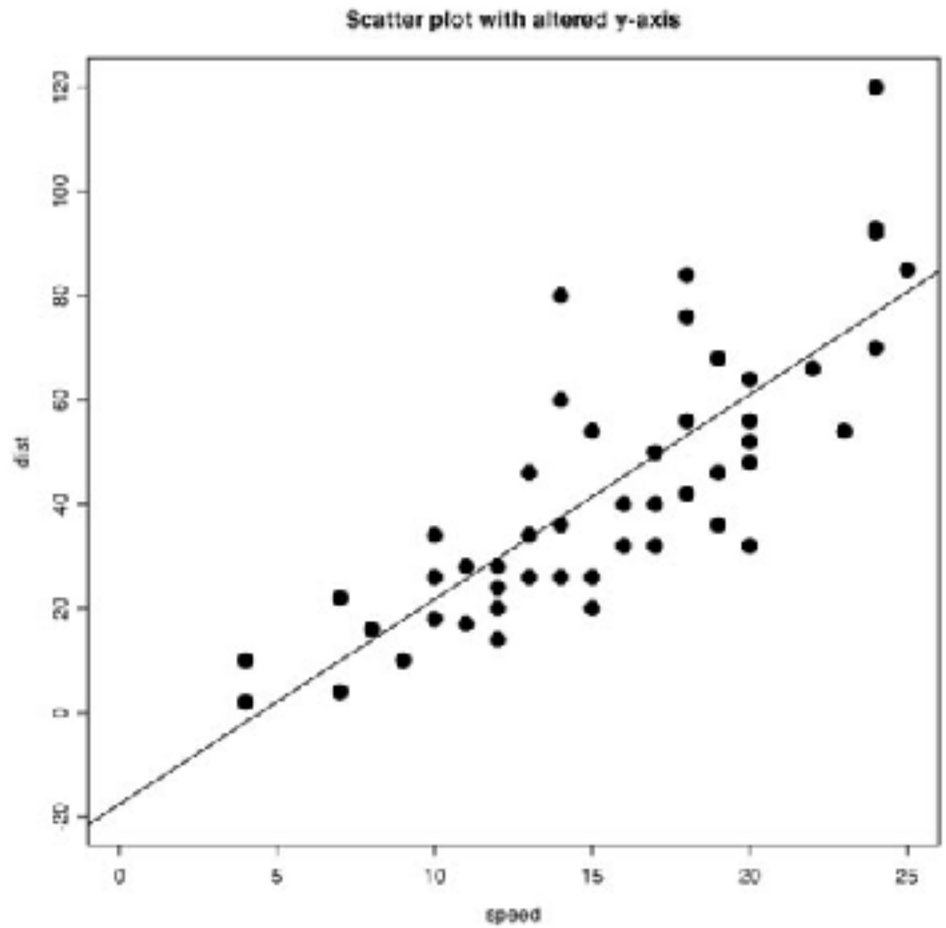
Dot Plots

- When to use:
  - When analyzing values that are spaced at irregular intervals
  - continuous, quantitative, univariate data



Scatter Plot

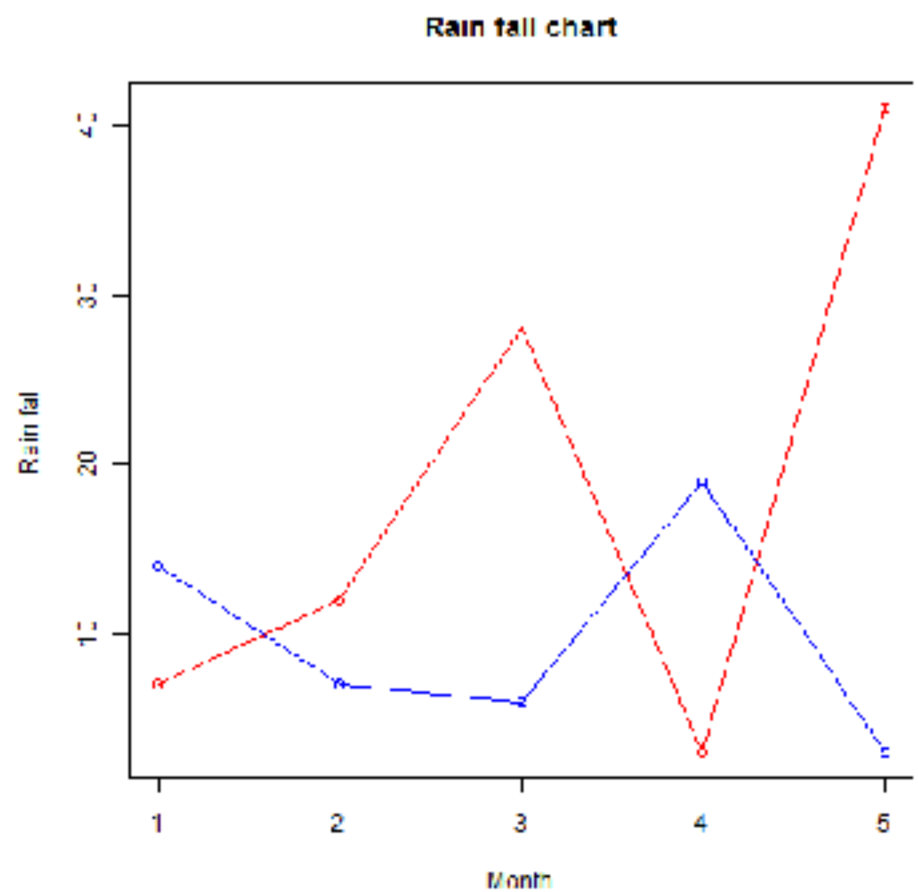
- When to use:
  - To compare how two quantitative variables change
  - continuous, quantitative, bivariate data
  - relationships for two variables



Line Graphs

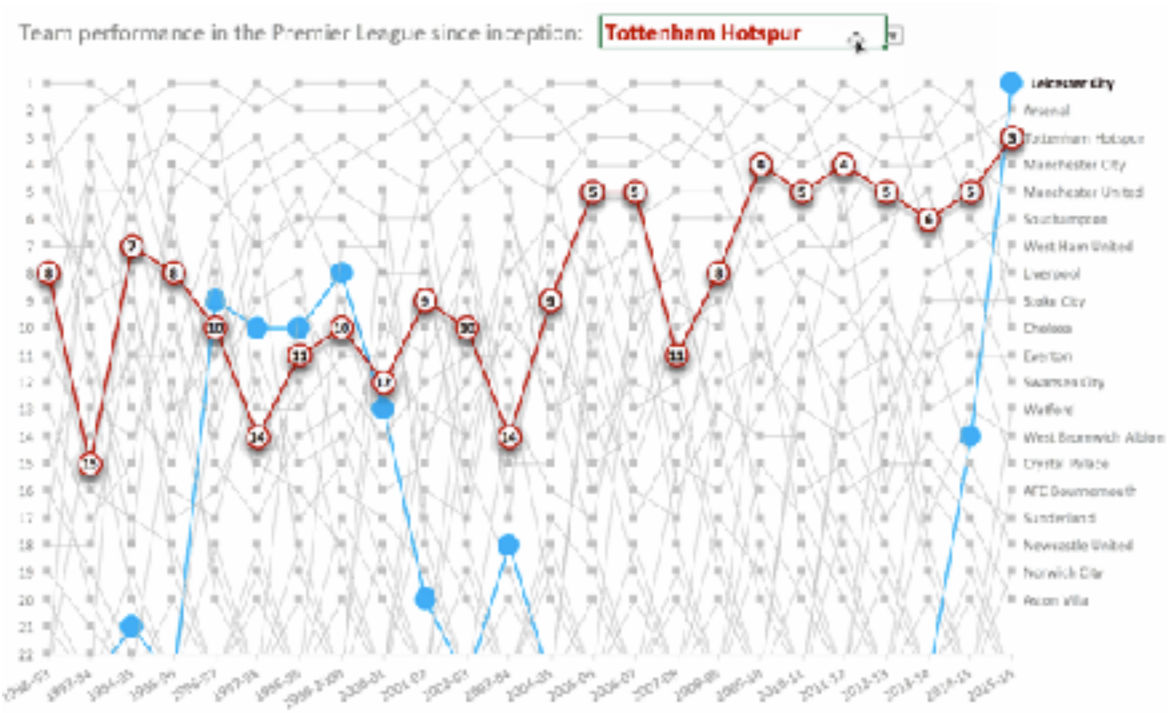
- When to use:
  – When quantitative values change during a continuous period of time (for more than one group)

  – Time series data (Non-cyclical data over time)



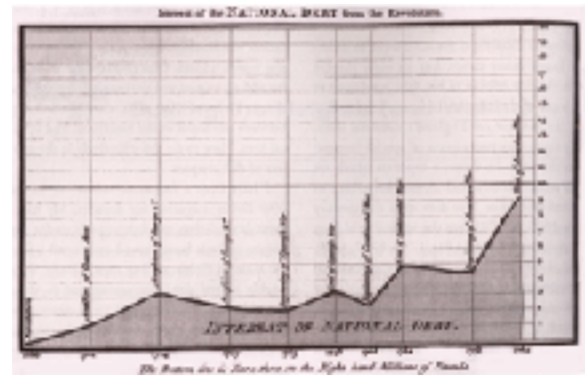
Bump Chart

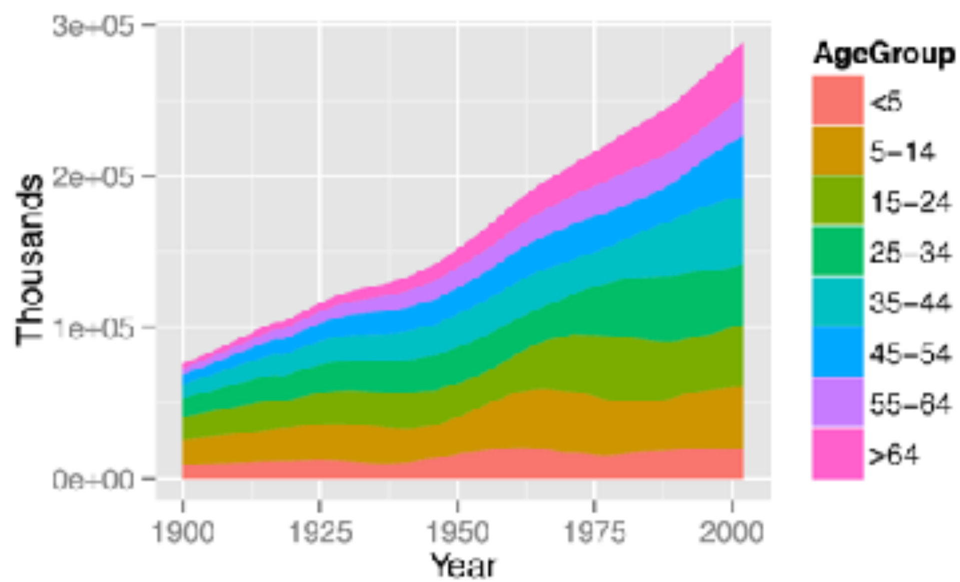
- When to use:
  - Similar to line graph
  - Y-axis: rank rather than (continuous) values



Area Graph

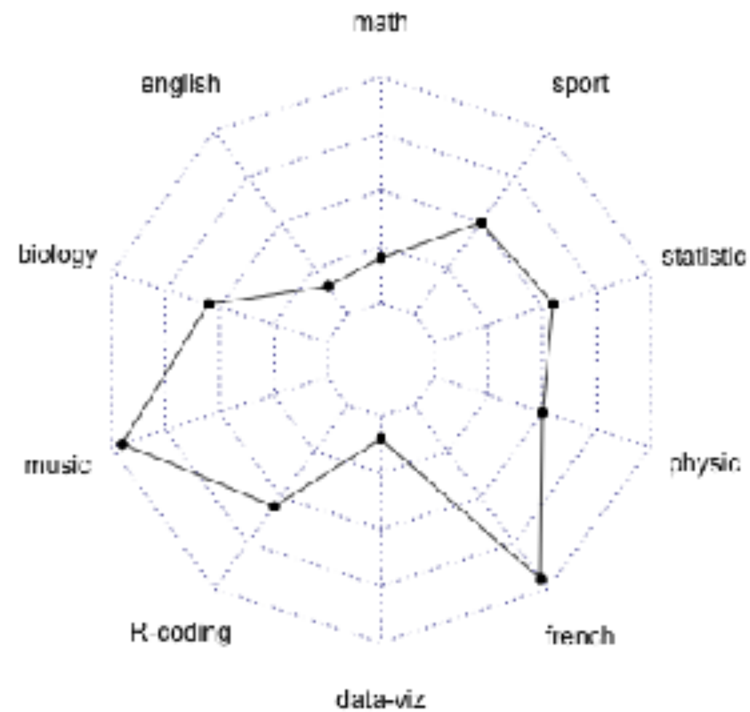
- When to use:
  - Commonly one compares with an area chart two or more quantities.
  - The area between axis and line are commonly emphasized with colors and textures.





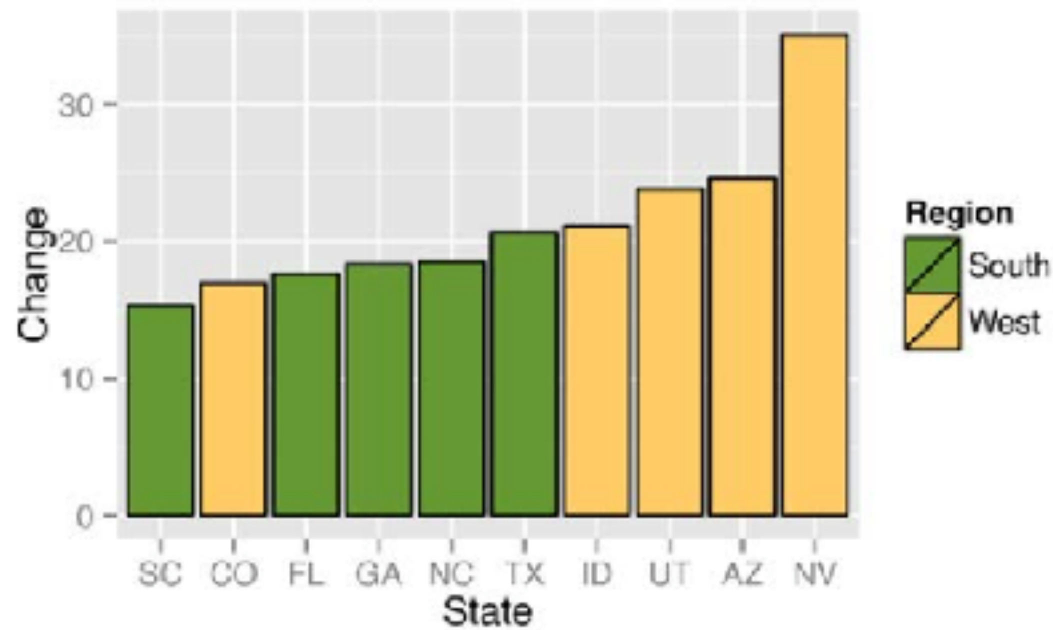
Radar Graphs

- When to use:
  - When you want to represent data across the cyclical nature of time
  - A two-dimensional chart of three or more quantitative variables represented on axes starting from the same point



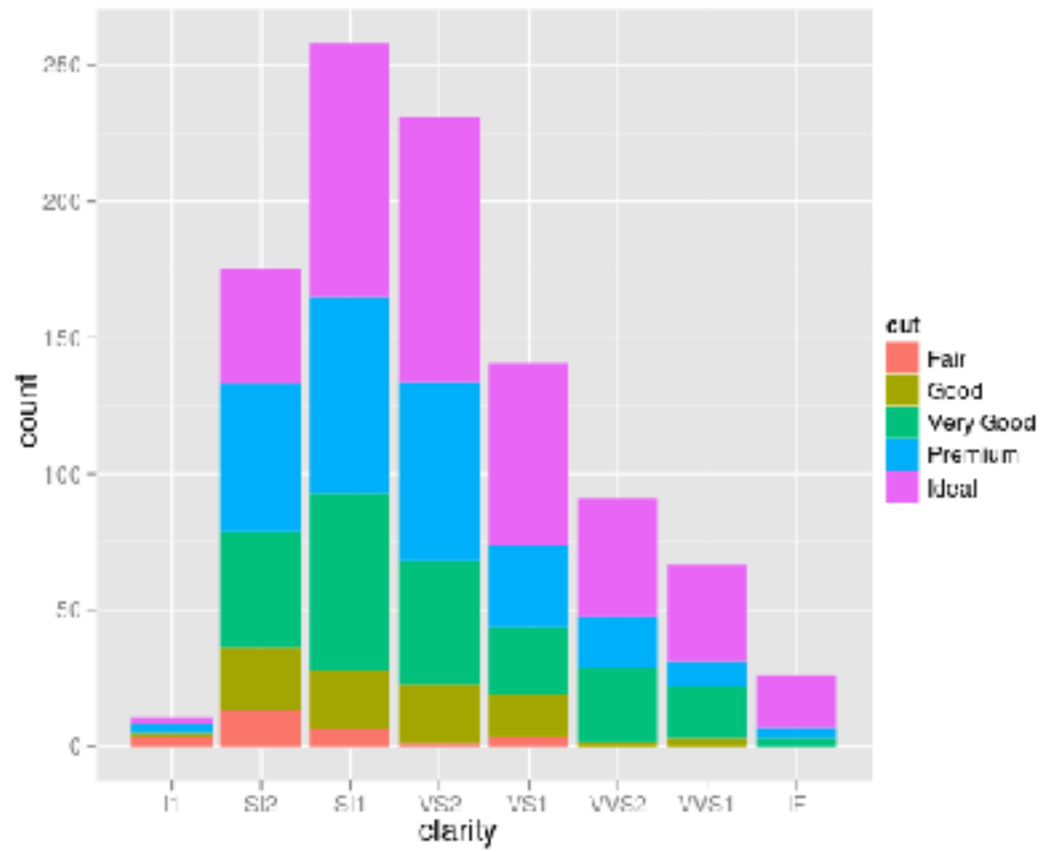
Bar Graphs

- When to use:
  - When you want to support the comparison of individual values between different groups
  - Can run vertically or horizontally



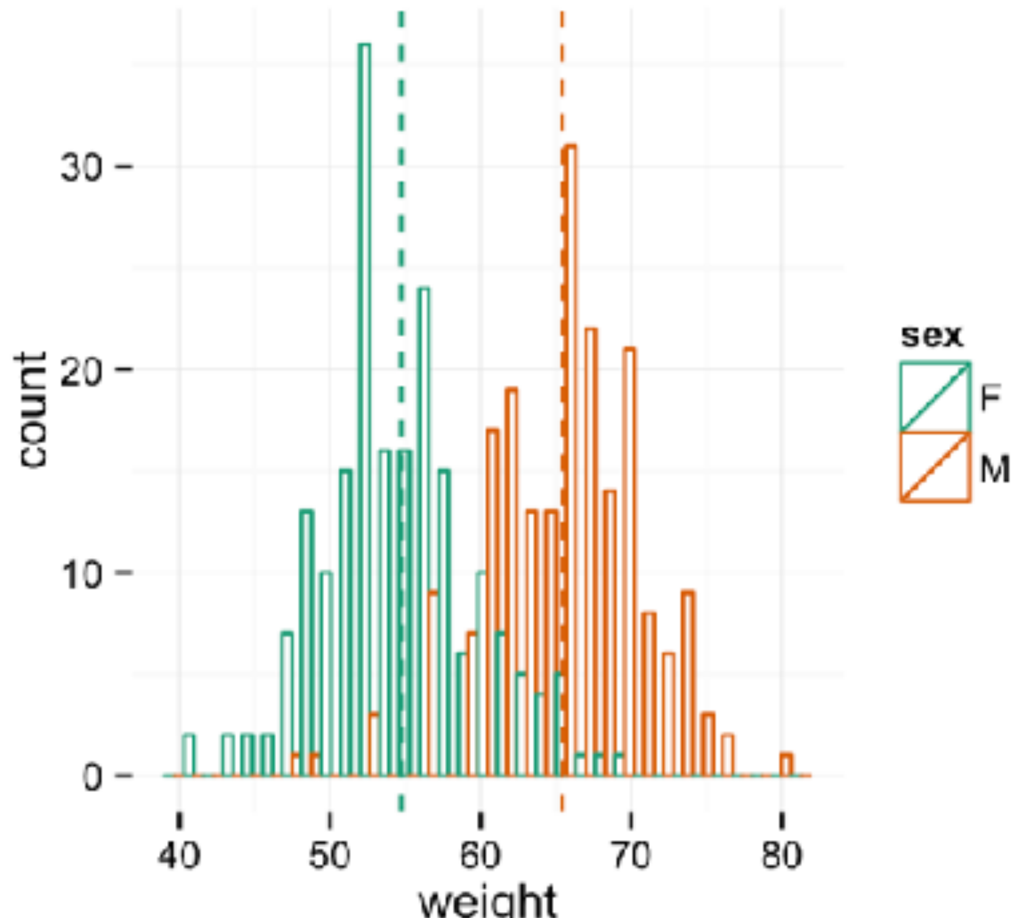
Stacked Bar Chart

- When to use:
  - When you want to present the total in a clear way while comparing part-to-whole relationship between different groups
  - Harder to compare the size of each categories



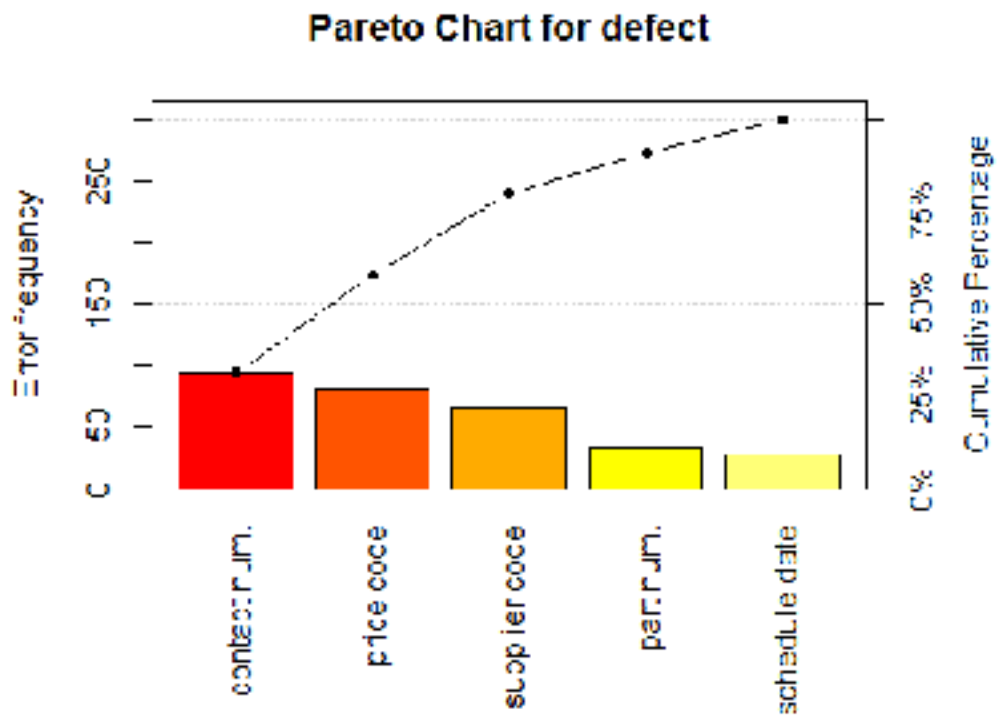
Histogram

- When to use:
  - the most commonly used graph to show frequency distributions
  - Continuous, quantitative, univariate data



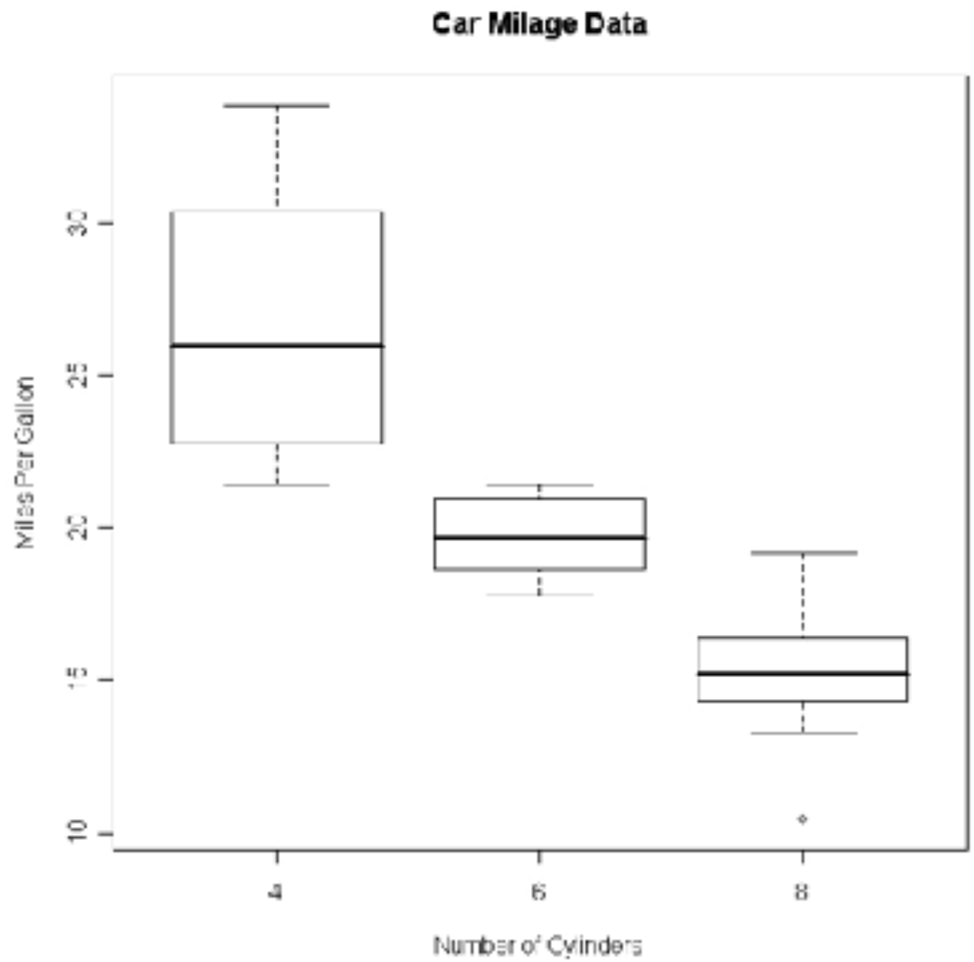
Pareto chart

- When to use:
  - When analyzing data about the frequency of problems or causes in a process.
  - containing both bars and a line graph



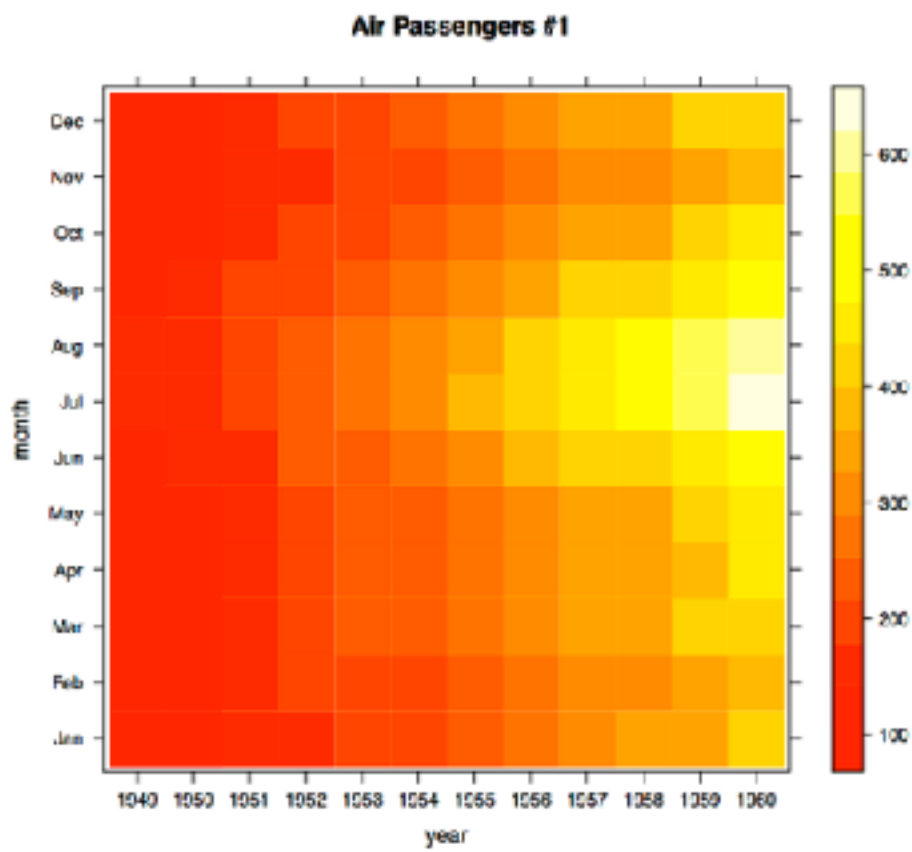
Box Plots

- When to use:
  - You want to show allow for comparison of data from different categories
  - graphically depicting groups of numerical data through their quartiles



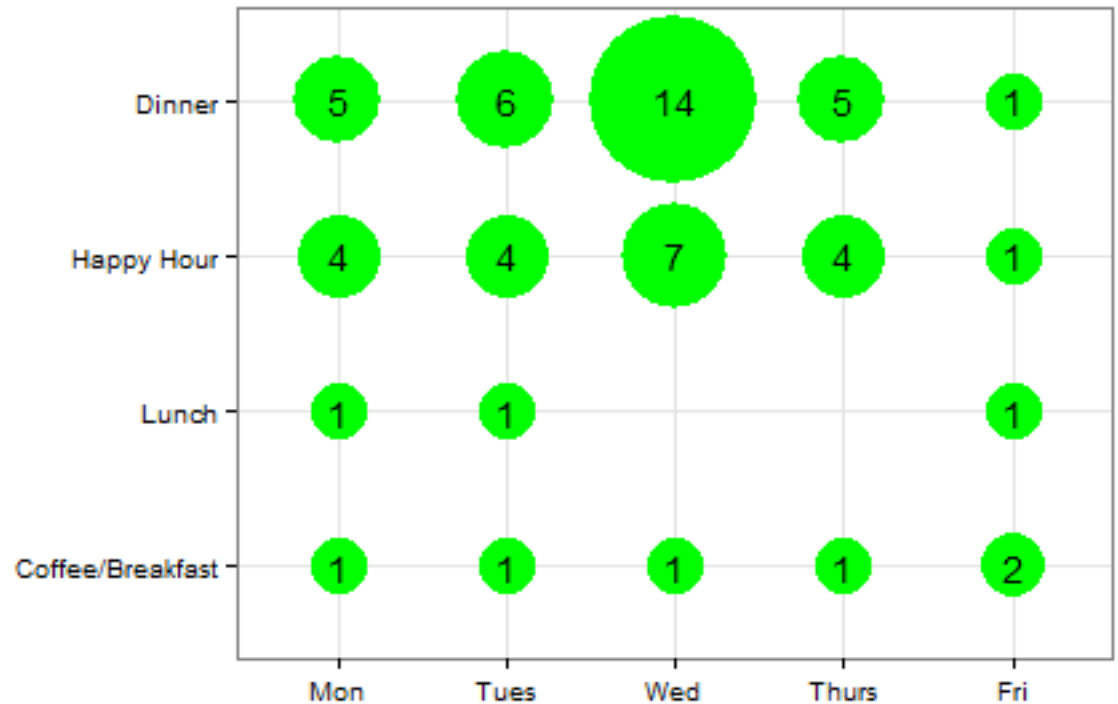
Heat Maps

- When to use:
  - When you want to display a large quantity of cyclical data (too much for radar)
  - Color choices: grayscales, rainbow, etc.



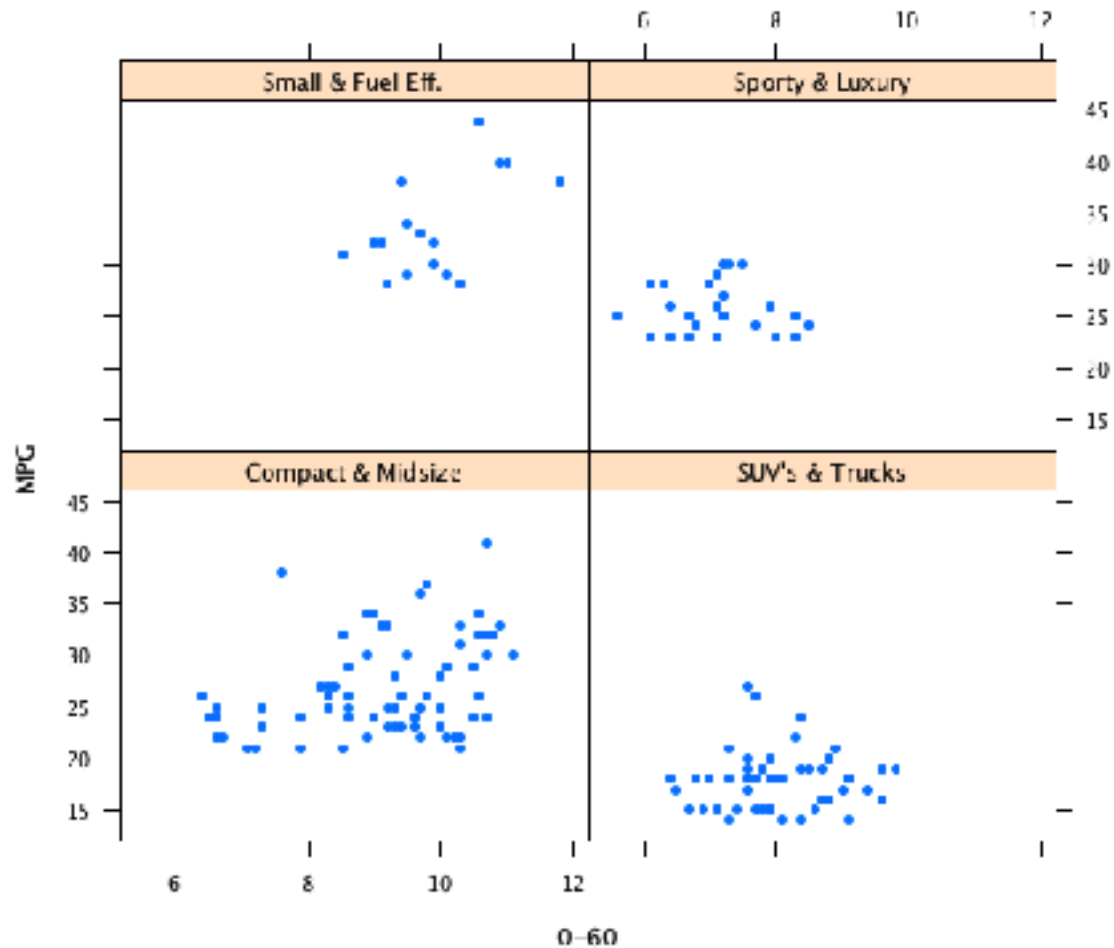
Crosstab Plot

- When to use:
  - Comparing different groups while presenting values (count)
  - Similar to heatmap



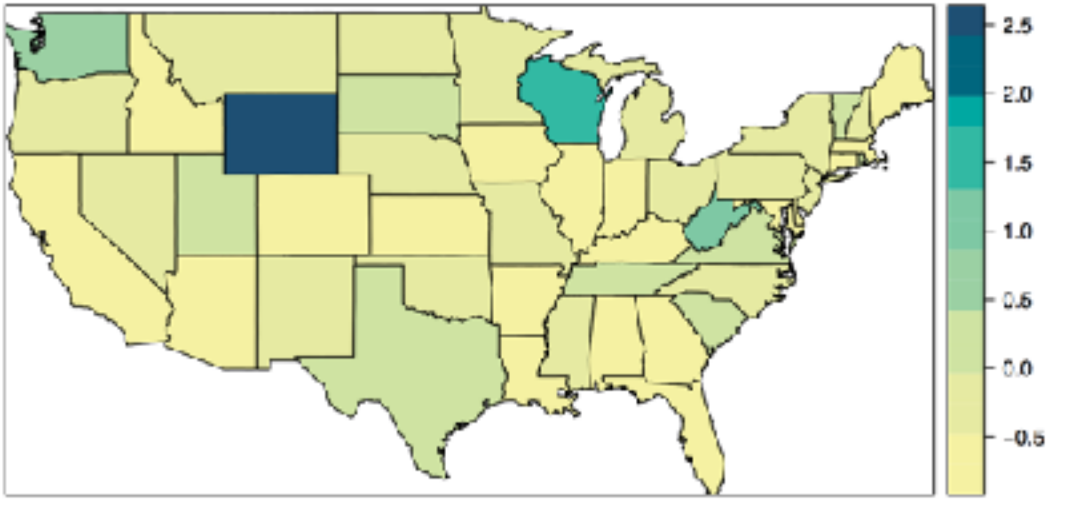
Trellis Display

- When to use:
  - Typically varies on one variable
  - Distribute different values of that variable across views

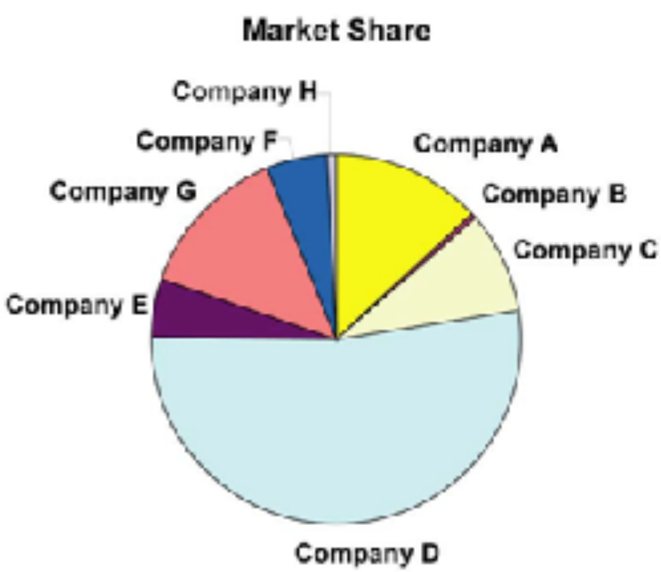


Hybrid: Map based Heatmap

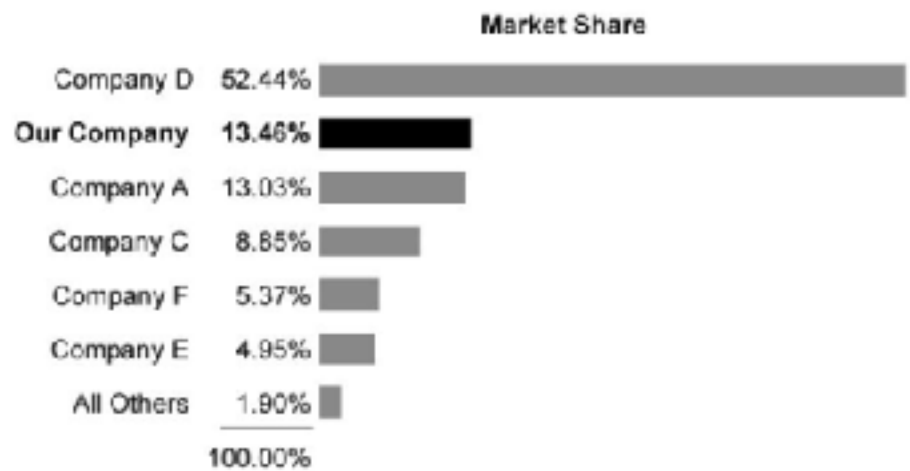
- When to use:
  - When you want to display a large quantity of cyclical data over different geo-locations



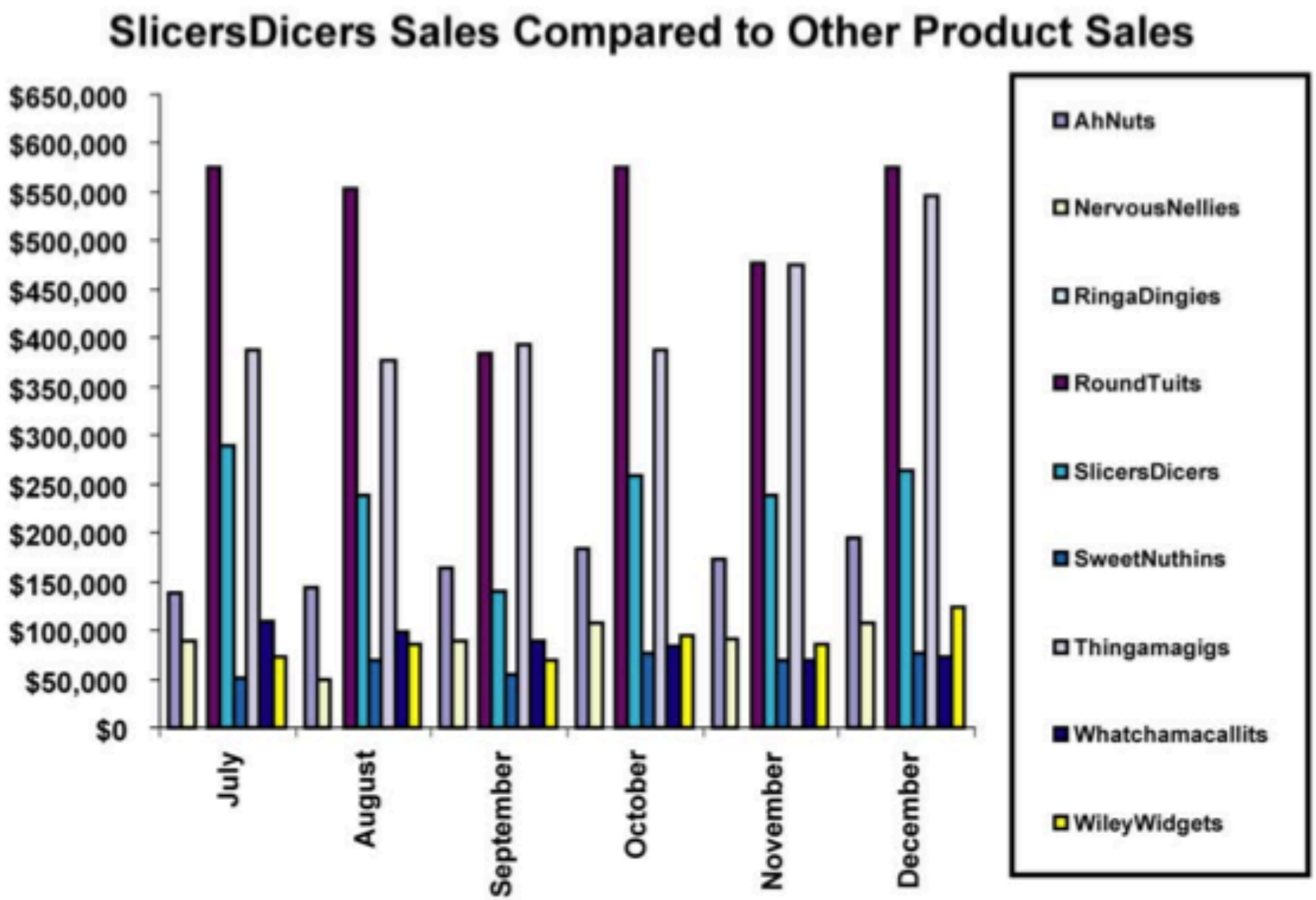
Comparisons



v.s.

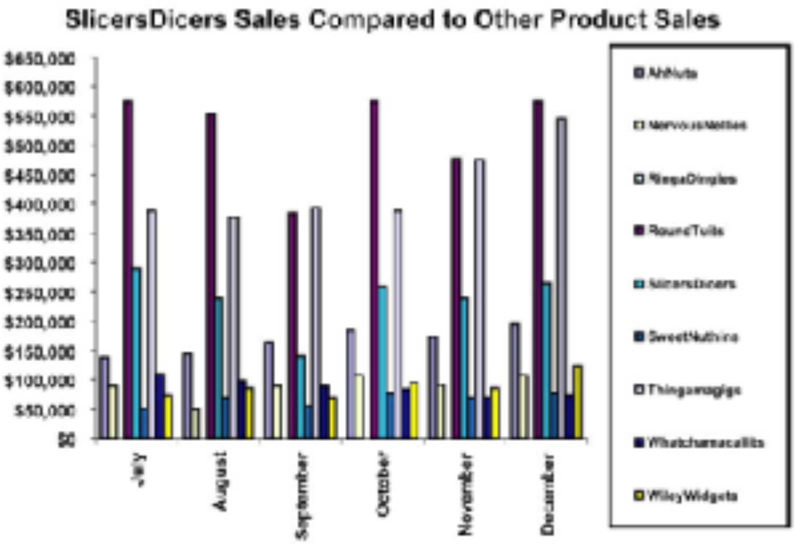


Comparisons

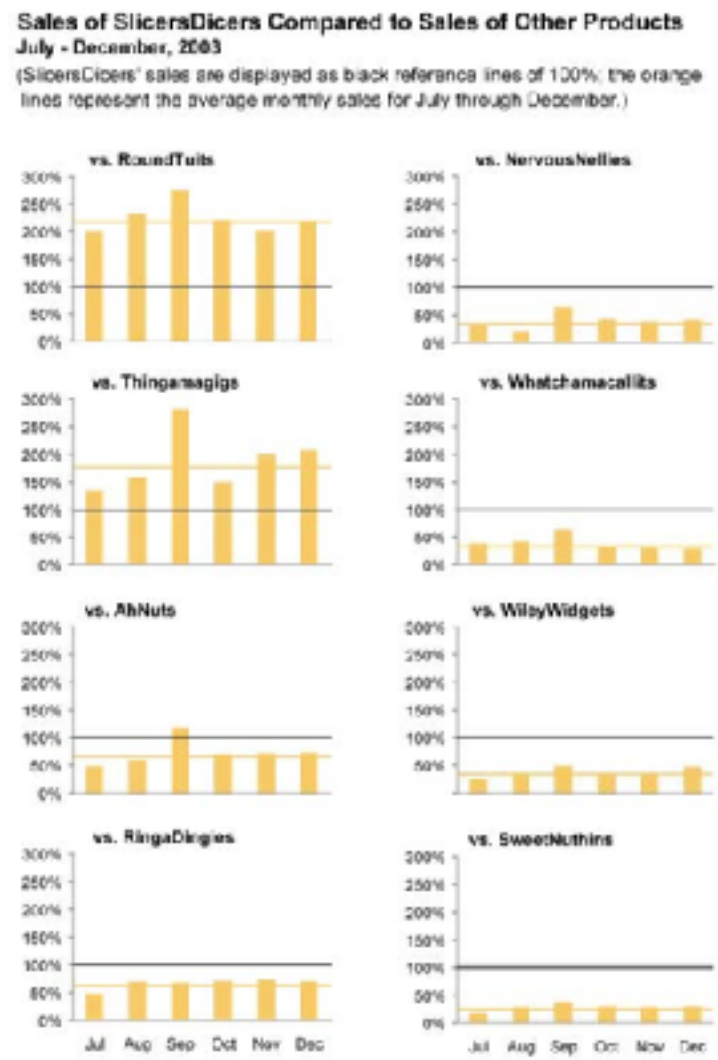


The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

Comparisons



v.s.



Next Lecture

- Topic: Multivariate Data Visualization
- Next Friday (21 Feb)
  - 13:00 - 15:00
  - A25, Business South, Jubilee Campus