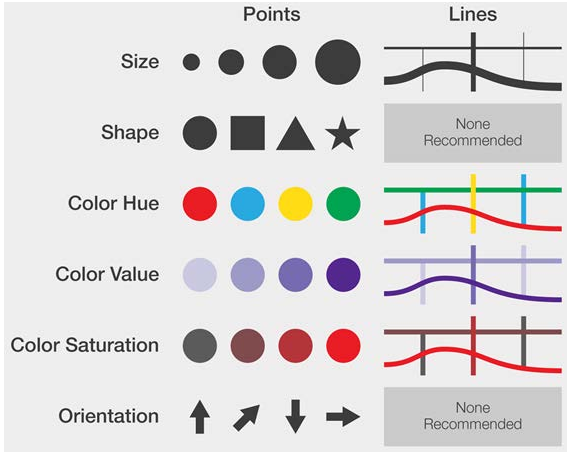
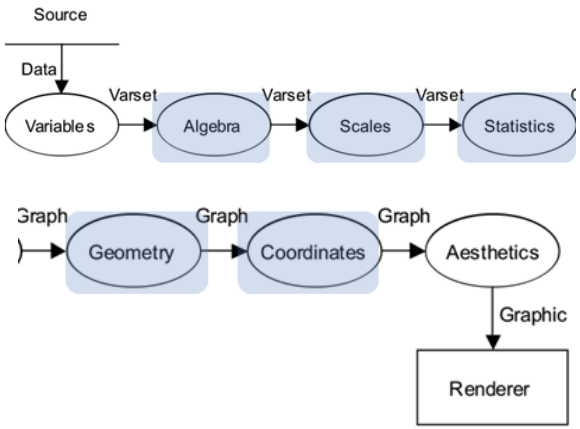




The Language of Graphs: from Bertin to GoG to ggplot2

Michael Friendly

Psych 6135

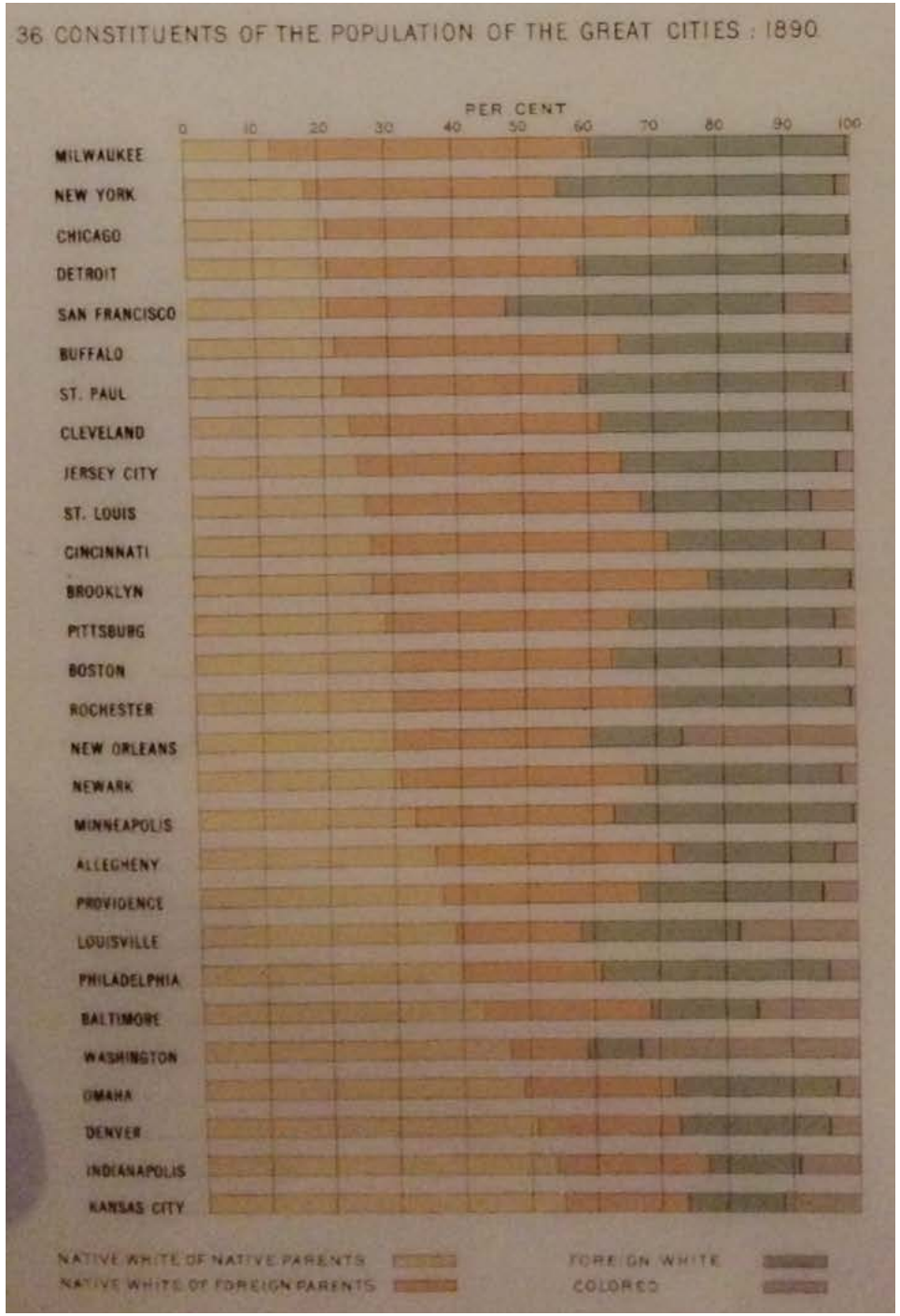
http://euclid.psych.yorku.ca/www/psy6135/

Topics

- Idea: Graphs as visual language
  - Early attempts at standardization of graphs
- Jacques Bertin: *Semiology of Graphics*
  - Mapping of visual properties to data relations
- Graphics programming languages:
  - Goal: power & elegance
- Lee Wilkinson: *Grammar of Graphics*
- Hadley Wickham: ggplot2

1870-1910: Statistical albums

- From ~ 1870—1910, statistical albums of official statistics on topics of population, trade, moral & political issues became widespread throughout Europe and the U.S.
  - France: *Album de Statistique Graphique*: 1879-1899 (trade, commerce & other topics)
  - USA: Census atlases: 1870/80/90--
    - Required by Congress to give proportional representation in the House of Representatives
    - Designed to give a "portrait of the nation"
  - Switzerland: Atlas graphique de la Suisse:1897, 1914
  - Others: Germany, Latvia, Romania, Bulgaria, etc.

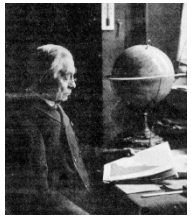

36 CONSTITUENTS OF THE POPULATION OF THE GREAT CITIES : 1890
PER CENT
0 10 20 30 40 50 60 70 80 90 100
MILWAUKEE
NEW YORK
CHICAGO
DETROIT
SAN FRANCISCO
BUFFALO
ST. PAUL
CLEVELAND
JERSEY CITY
ST. LOUIS
CINCINNATI
BROOKLYN
PITTSBURG
BOSTON
ROCHESTER
NEW ORLEANS
NEWARK
MINNEAPOLIS
ALLEGHENY
PROVIDENCE
LOUISVILLE
PHILADELPHIA
BALTIMORE
WASHINGTON
OMAHA
DENVER
INDIANAPOLIS
KANSAS CITY
NATIVE WHITE OF NATIVE PARENTS
NATIVE WHITE OF FOREIGN PARENTS
FOREIGN WHITE
COLORED


NATIVE WHITE OF NATIVE PARENTS
NATIVE WHITE OF FOREIGN PARENTS
FOREIGN WHITE
TOTAL COLORED
36 CONSTITUENTS OF THE POPULATION OF THE GREAT CITIES : 1890
39 NATIONALITIES OF THE FOREIGN BORN AT EACH CENSUS : 1850 to 1890

Need for standardization

- Beautiful graphics: Yes, but all separate designs
  - Can anything be compared across countries?
- Émile Cheysson (1878)
  - *"The time will come when Science has to lay down general principles and decide on well-defined standards. We can no longer tolerate this sort of anarchy"*
- International statistical meetings
  - 1852 (Brussels), 1857 (Vienna), 1869 (The Hague), 1872 (St. Petersburg), 1876 (Budapest) …
  - Participants: Cheysson, Levasseur (France), Ernest Engel, Gustav von Mayr, Hans Schwabe (Germany), Francis Walker (U.S.), …

Cheysson



Levasseur

von Mayr

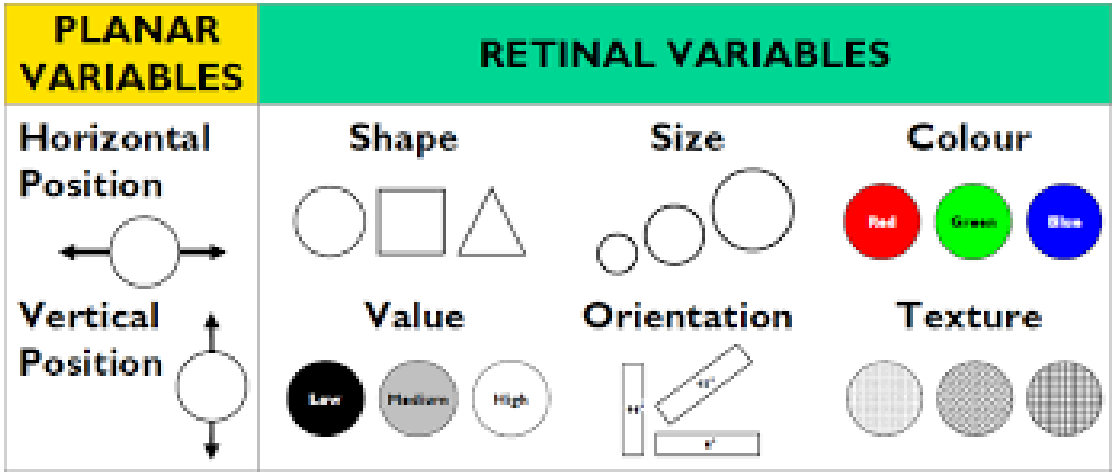
Walker

No consensus

- St. Petersburg (1872) resolutions:
  - "*The Congress accepts that it is not worth going into details about the choice of methods or facts for graphical representation*".
  - "*no strict rule can be imposed on authors, because the only real problem is that of applying the graphical method to data that is comparable*".
- Most of the debate had to do with thematic maps
  - number of class intervals for a quantitative variable
  - number and variety of shading colors
- Yet, the idea of a visual language had been accepted, along with the need for some theory of graphs

Bertin: *Semiology of graphics* (1967)

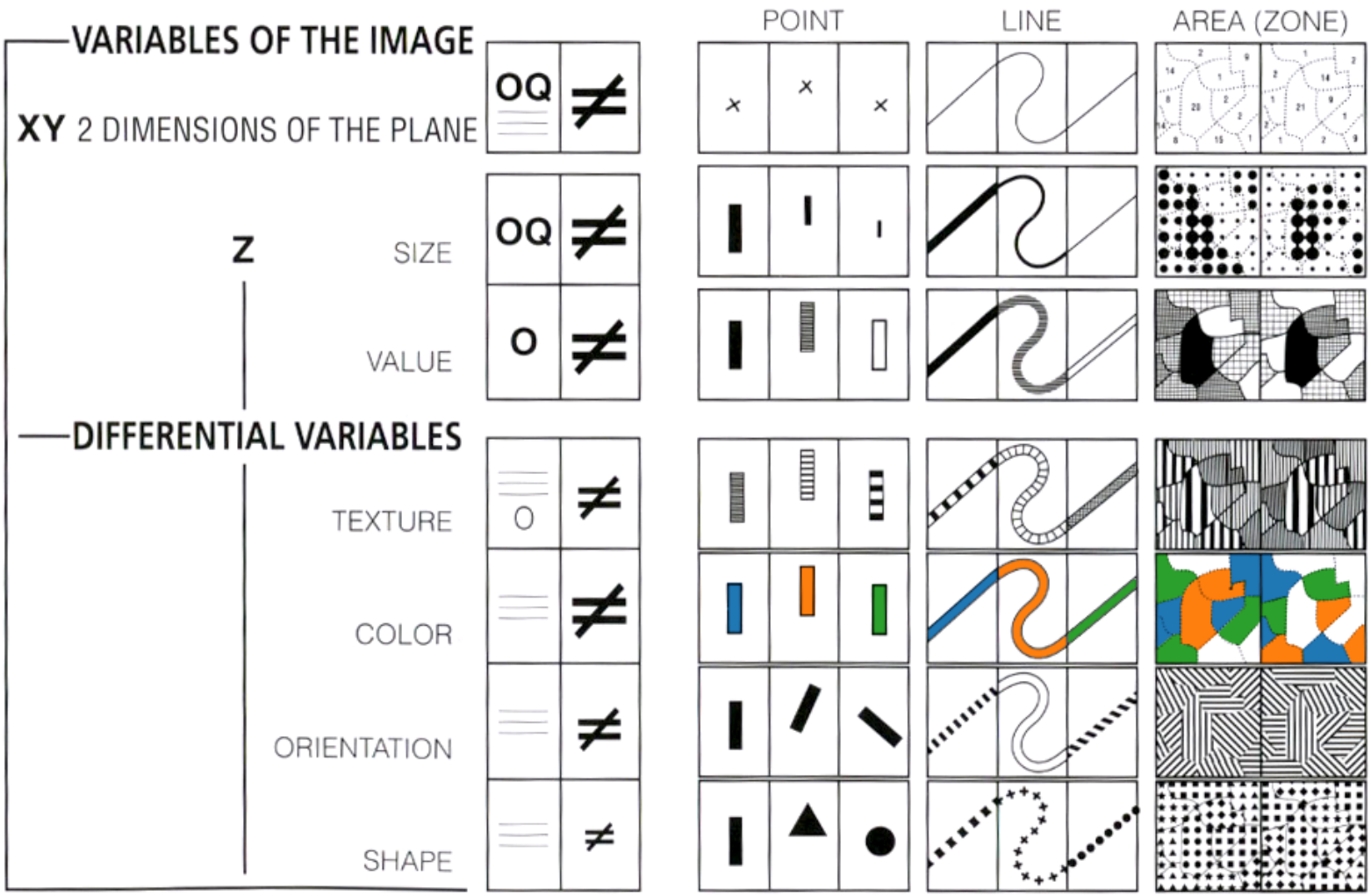
- Defines a system of "grammatical elements" of graphs and relations among visual attributes that give meaning (semantics) from perceptual features
  - Planar variables: (x,y) coordinates
  - Retinal variables: shape, size, color, …



Bertin: *Semiology of graphics*

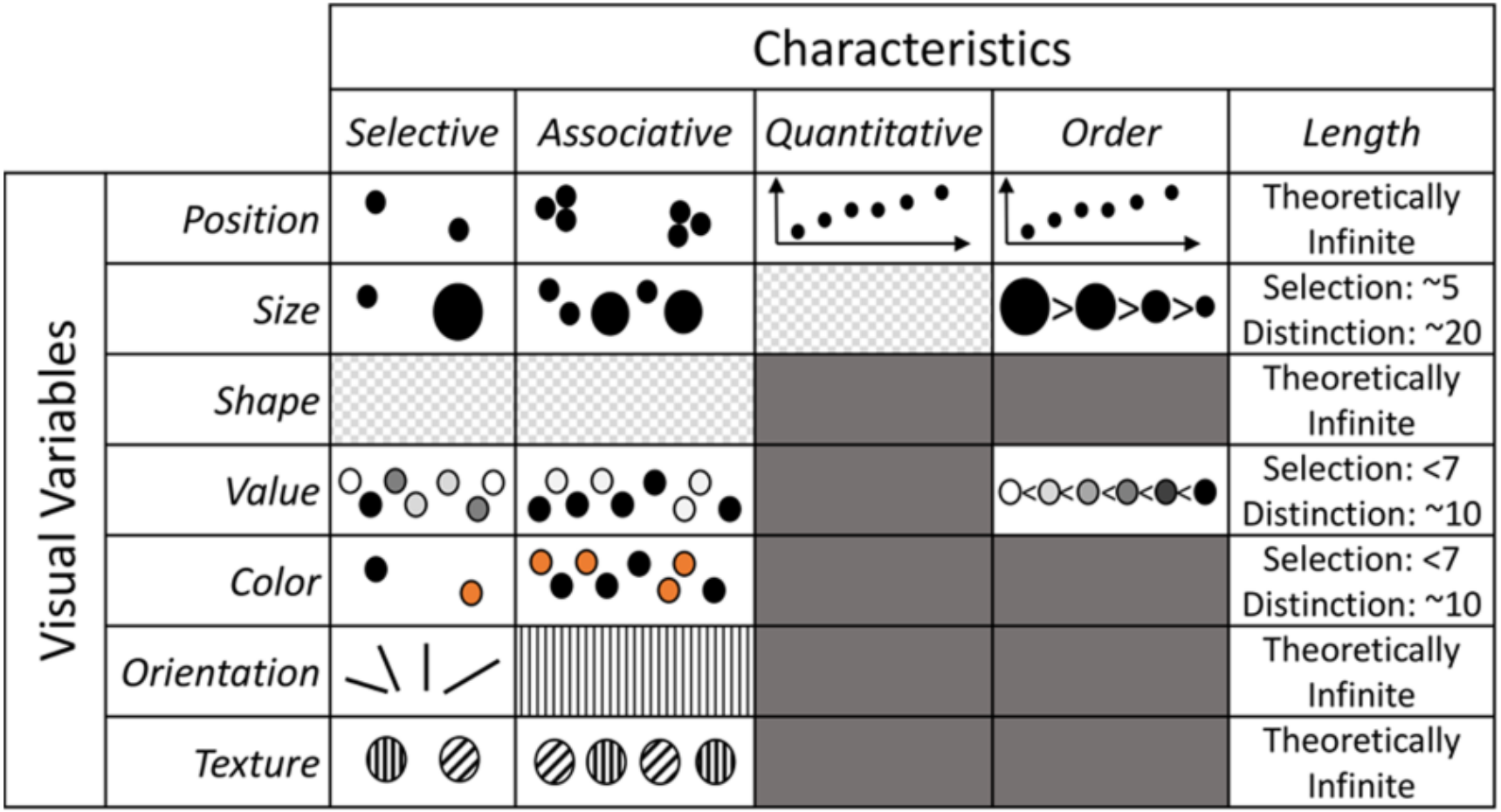
- Defines a system of mapping of retinal variables to properties of data variables for perception of relations
  - Association (≡) – marks are perceived as similar
  - Selection (≠) – marks are perceived as forming classes
  - Order (O) – marks are perceived as showing order
  - Quantity (Q) – marks are perceived as proportional
- This is the first theory of graphs relating visual attributes (encoding) to perceptual characteristics (decoding).
- It comprises nearly all known graph and thematic map types in a general system

The retinal variables and relationship types can be implanted in various symbol types in the plane (X,Y)



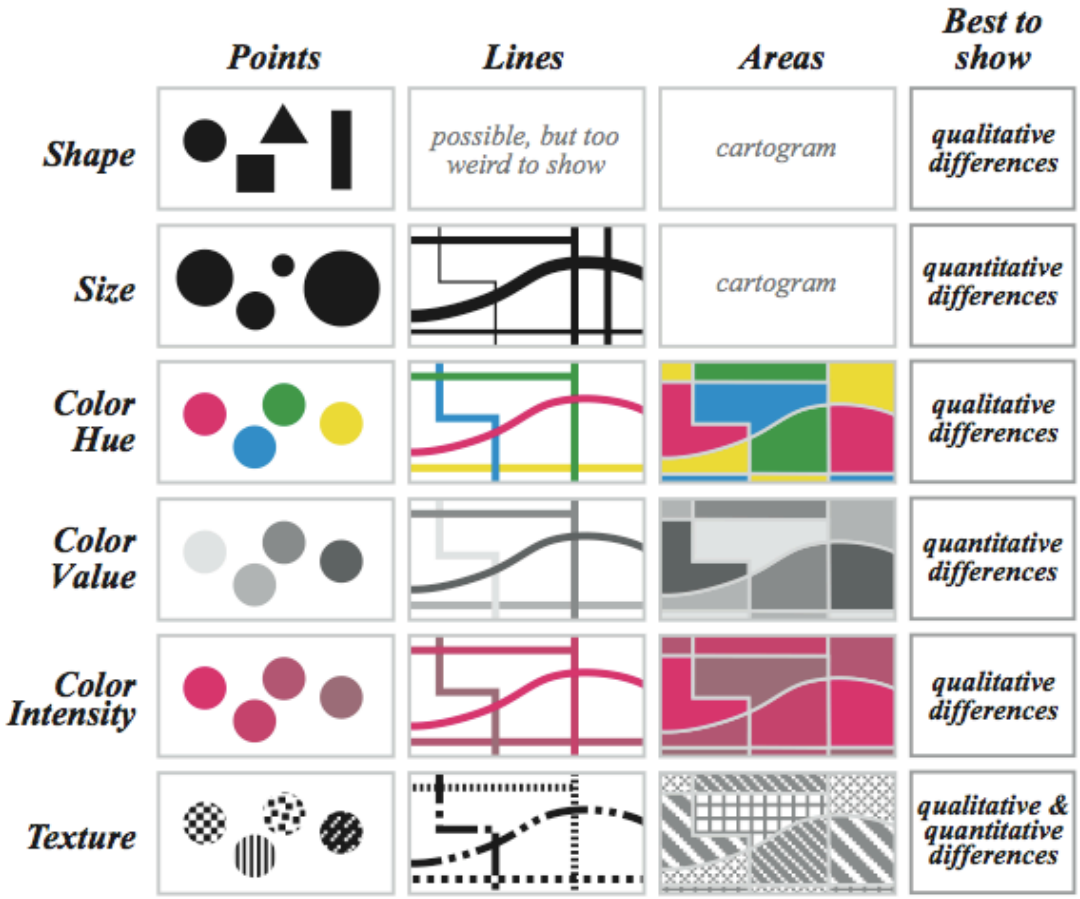
Visual variables & data characteristics

Visual variables differ in the kinds of information they can convey



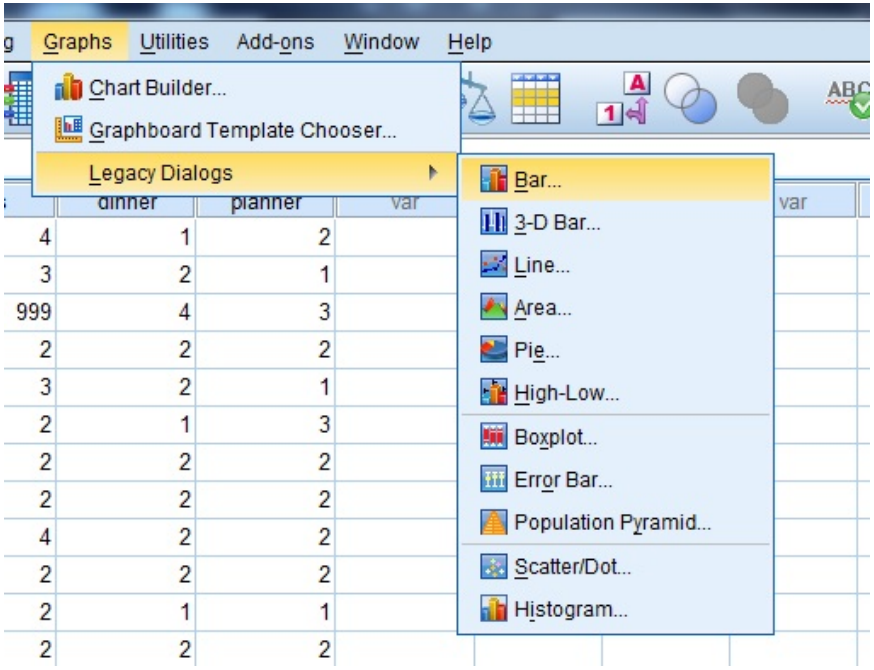
Some recommendations

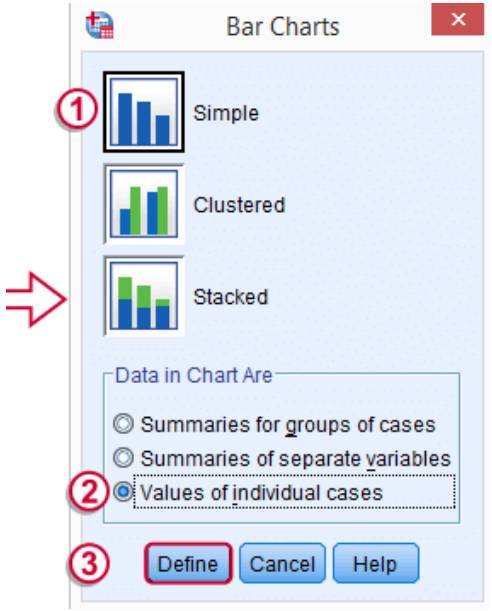
Various authors have used Bertin's system to make recommendations for the best attributes to use with different symbol types



Making graphs: menus vs. syntax

Menu-driven graphics provide a wide range of graph types, with options
What's wrong with that?



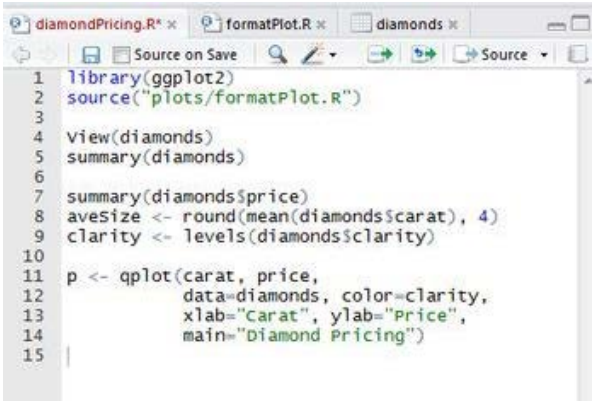


WYSIAYG: What you see is **all** you get. No way to do something different
Not reproducible: Change the data → Re-do manually from scratch
Often designed by programmers with little sense of data vis

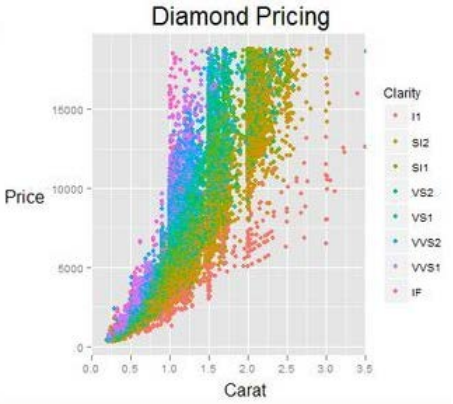
Programming languages: Power & elegance

- **CS view**: All programming languages can be proved to be equivalent (to a Turing machine)
- **Cognitive view**: Languages differ in:
  - expressive power: ease of translating what you want to do into the results you want
  - elegance: how well does the code provide a human-readable description of what is done?
  - extensibility: ease of generalizing a method to wider scope
  - learn-ability: your learning curve (rate, asymptote)

+



→

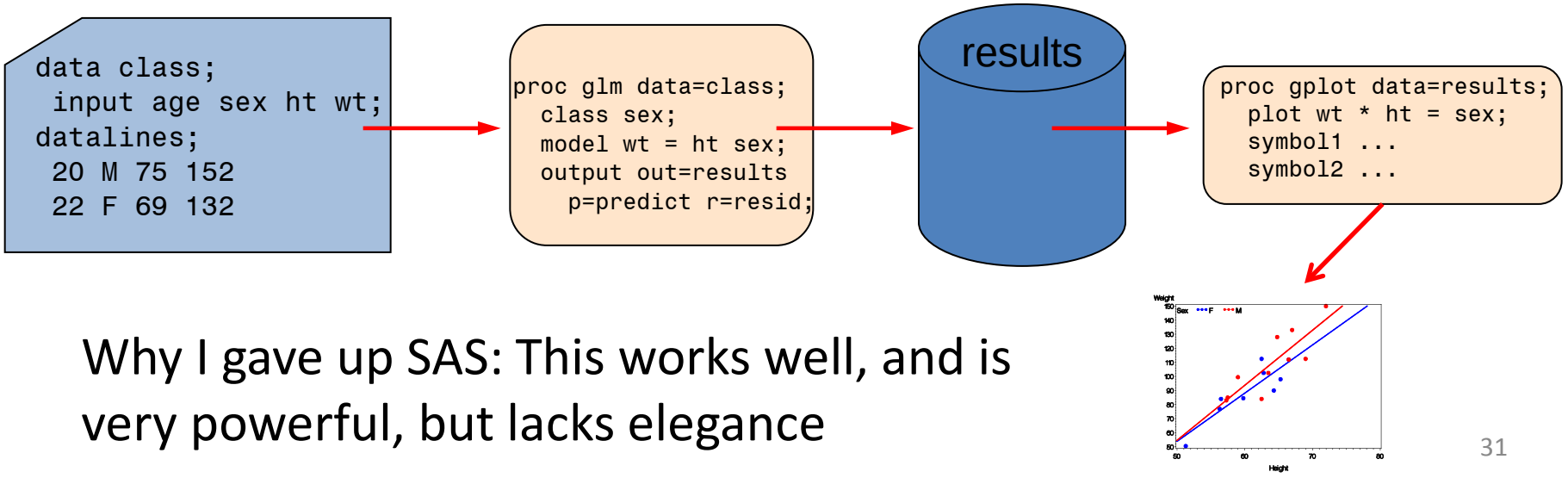


Programming languages: Power & elegance

Language	Features:Tools for thinking?
FORTRAN	Subroutines – reusable code Subroutine libraries (e.g., BLAS)
APL, *APL2STAT*	N-way arrays, nested arrays Generalized reduction, outer product Function operators
Logo	Turtle graphics Recursion, list processing
Lisp, LispStat, *ViSta*	Object-oriented computing Functional programming
Perl	Regular expressions Search, match, transform, apply
SAS	Data steps, PROC steps, BY processing SAS macros, Output Delivery system
R	Object-oriented methods, tidyverse: dplyr, ggplot2, …

Graphics programming languages: SAS

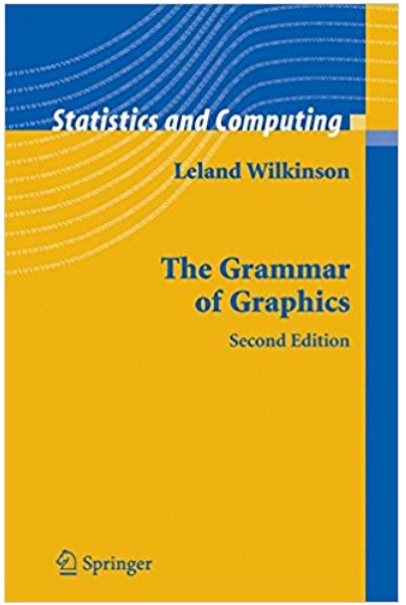
- SAS: procedures + annotate facility + macros
  - PROC GPLOT (x,y plots), PROC GCHART, PROC GMAP, …
  - Annotate: data set with instructions (move, draw, text, fonts, colors)
  - Macros: Create a new, generic plot type, combining PROC steps and DATA steps.



Why I gave up SAS: This works well, and is very powerful, but lacks elegance

Wilkinson: Grammar of Graphics

- Natural language:
  - Grammar/syntax: What are the minimal, complete set of rules to describe all well-formed sentences?
    - John ate the big red apple
    - John big apple red apple ate the
  - Semantics: How to distinguish meaning, nonsense, poetry in well-formed sentences?
    - Large green trucks carry garbage
    - Colorless green ideas sleep furiously
- How to apply these ideas to graphics?
  - Grammar: Algebra, scales, statistics, geometry, …
  - Semantics: Space, time, uncertainty, …
  - Needed: a complete formal theory of graphs & computational graphics language

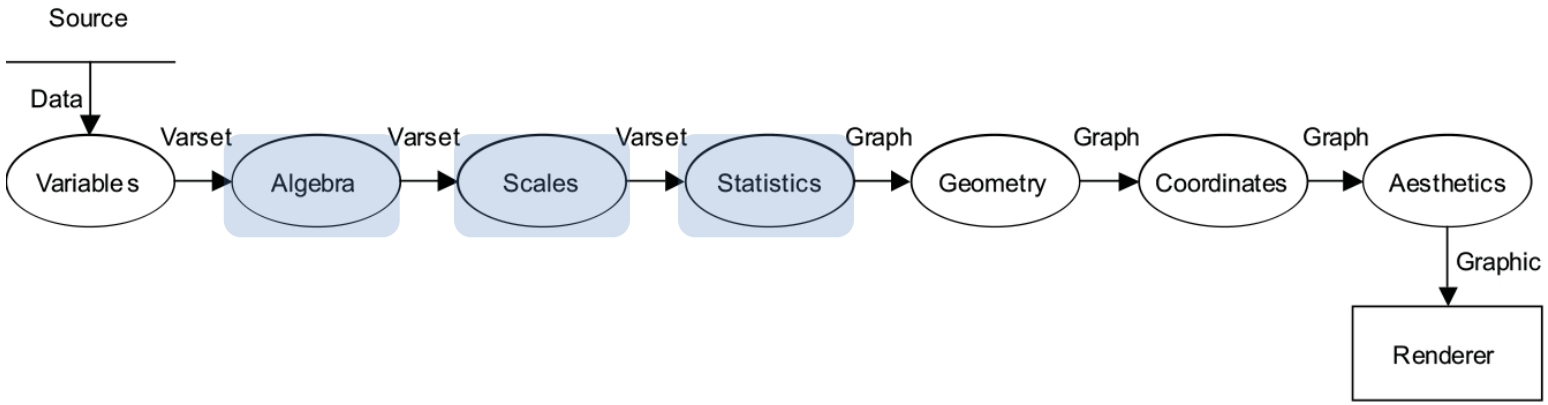


Wilkinson: Grammar of Graphics

- A complete system, describing the components of graphs and how they combine to produce a finished graphic
  - *"The grammar of graphics takes us beyond a limited set of charts (words) to an almost unlimited world of graphical forms (statements)"* (Wilkinson, 2005, p. 1).
  - *"... describes the meaning of what we do when we construct statistical graphics … more than a taxonomy"*
  - *"This system is capable of producing some hideous graphics … This system cannot produce a meaningless graphic, however."*
- This is a general theory for producing graphs.
  - the foundation of most modern software systems;
  - not connected with a theory for reading graphs à la Bertin.

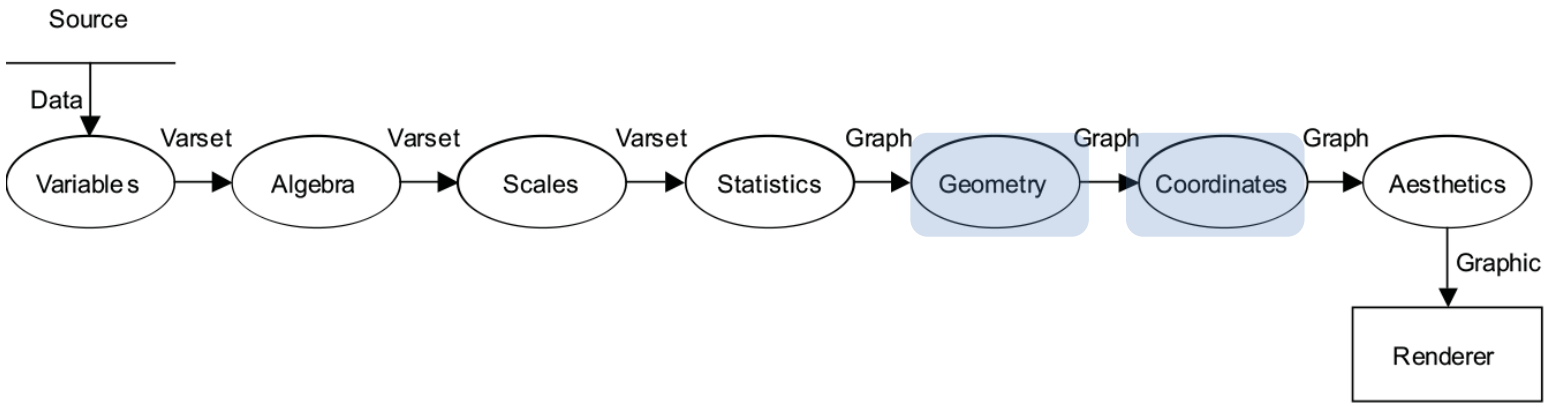
Grammar of Graphics: Specification

- **Algebra**: combine variables into a data set to be plotted
  - cross (A*B), nest (A/B), blend (A+B), filter, subset, …
- **Scales**: how variables are represented
  - categorical, linear, log, power, logit, …
- **Statistics**: computations on the data
  - binning, summary (mean, median, sd), region (CI), smoothing



Grammar of Graphics: Specification

- **Geometry**: Creation of geometric objects from variables
  - Functions: point, line, area, interval, path, …
  - Partitions: polygon, contour,
  - Networks: edge
  - Collision modifiers: stack, dodge, jitter
- **Coordinates**: Coordinate system for plotting
  - transformations: translation, rotation, dilation, shear, projection
  - mappings: Cartesian, polar, map projections, warping, Barycentric
  - 3D+: spherical, cylindrical, dimension reduction (MDS, SVD, PCA)

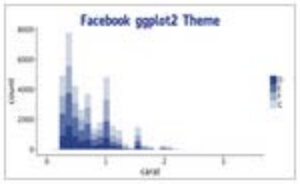


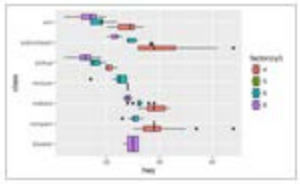
Grammar of Graphics: Specification

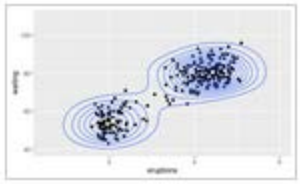
- **Aesthetics**: mapping of qualitative and quantitative scales to sensory attributes (extends Bertin)
  - Form: position, size, shape (polygon, glyph, image), rotation, …
  - Surface: color (hue, saturation, brightness), texture (pattern, orientation), blur, transparency
  - Motion: direction, speed, acceleration
  - Sound: tone, volume, rhythm, voice, …
  - Text: label, font, size, …
- **Facets**: Construct multiplots (“small multiples”) by partitioning, blending or nesting
- **Guides**: Allow for reading the encodings of variables mapped to aesthetics
  - scales: axes, legend (labels: size, shape, color, …)
  - annotations (title, footnote, line, arrow, ellipse, text, …)

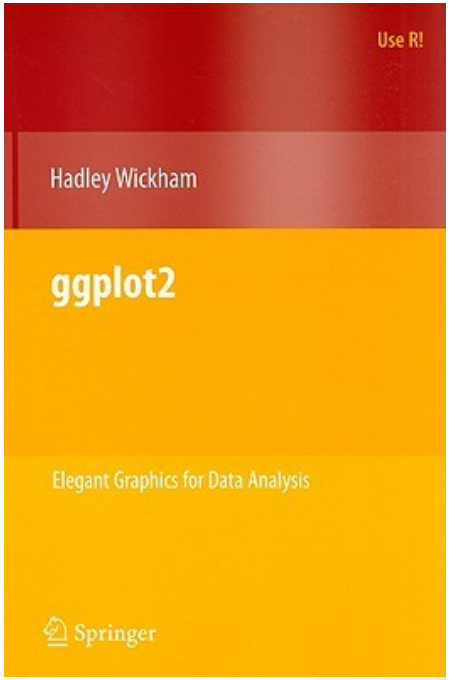
Wickham: ggplot2

- ggplot2: *Elegant graphics for data analysis*
  - a computational language for thinking about & constructing graphs
  - sensible, aesthetically pleasing defaults
    - + themes: default, bw, journal, tufte, …
  - infinitely extendable
    - ggplot extensions: http://www.ggplot2-exts.org/







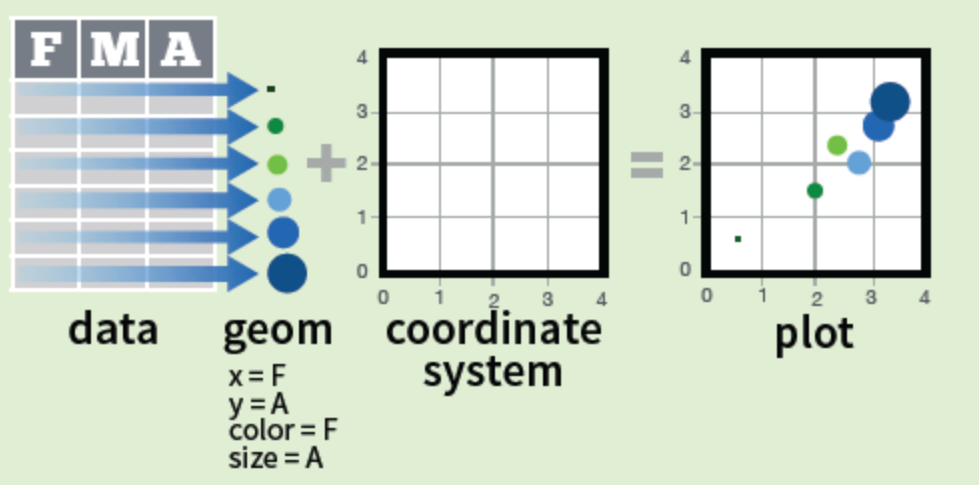


Wickham: ggplot2

- Implementation of GoG in R as layers of a graphic
  - Basic layers:
    - Data,
    - Aesthetics (data → plot mapping)
    - Geoms (points, lines, bars, …),
  - Statistics: summaries & models
  - Coordinates: plotting space
  - Facets: partition into sub-plots
  - Themes: define the general features of all graphical elements

ggplot2: data + geom = graph

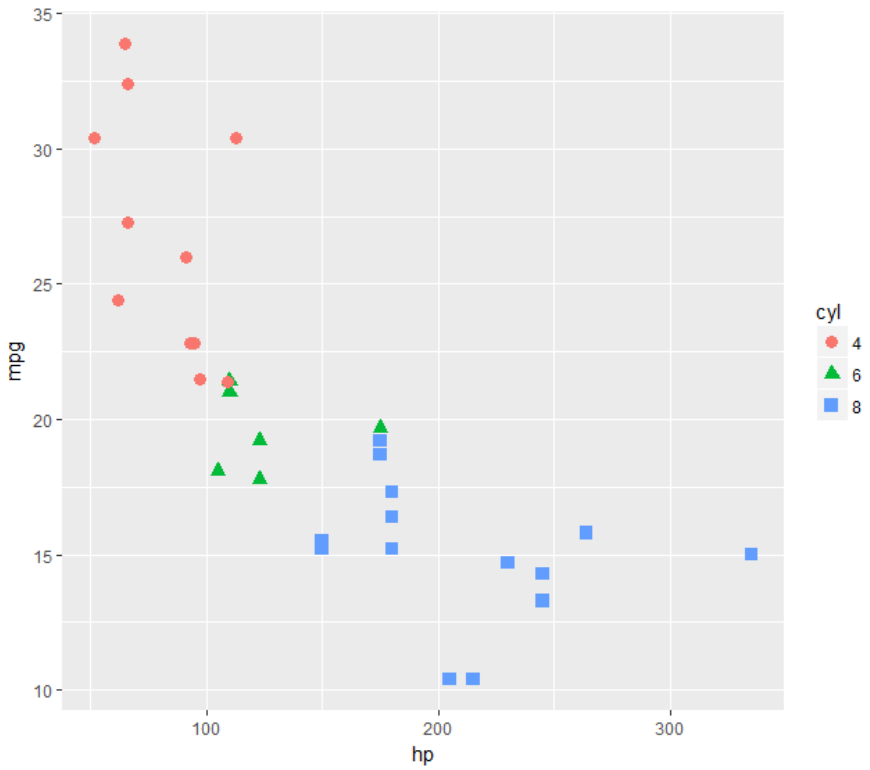
- Every graph can be described as a combination of independent building blocks, connected by “+” (read: “and”)
  - data: a data frame: quantitative, categorical; local or data base query
  - aesthetic mapping of variables into visual properties: size, color, x, y
  - geometric objects (“geom”): points, lines, areas, arrows, …
  - coordinate system (“coord”): Cartesian, log, polar, map,




```
ggplot(FMA,
          aes(x=F, y=A, color=F, size=A) +
  geom_point()
```

ggplot2: data + geom = graph

```
ggplot(data=mtcars,
        aes(x=hp, y=mpg,
            color=cyl, shape=cyl)) +
    geom_point(size=3)
```



In this call:

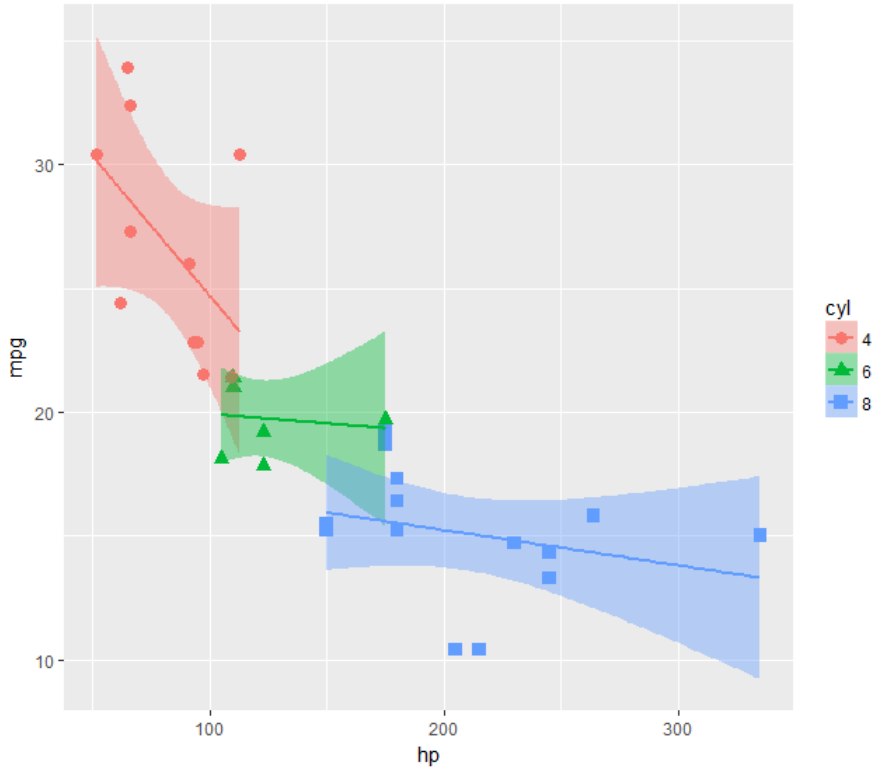
- data=mtcars: data frame
- aes(x=, y=): plot X,Y variables
- aes(color=, shape=): attributes
- + geom_point(): what to plot
- the coordinate system is taken to be the standard Cartesian (x,y)
- a corresponding legend is automatically generated

ggplot2: geoms

Wow! I can really see something there.

How can I enhance this visualization?

Easy: add a geom_smooth() to fit linear regressions for each level of cyl



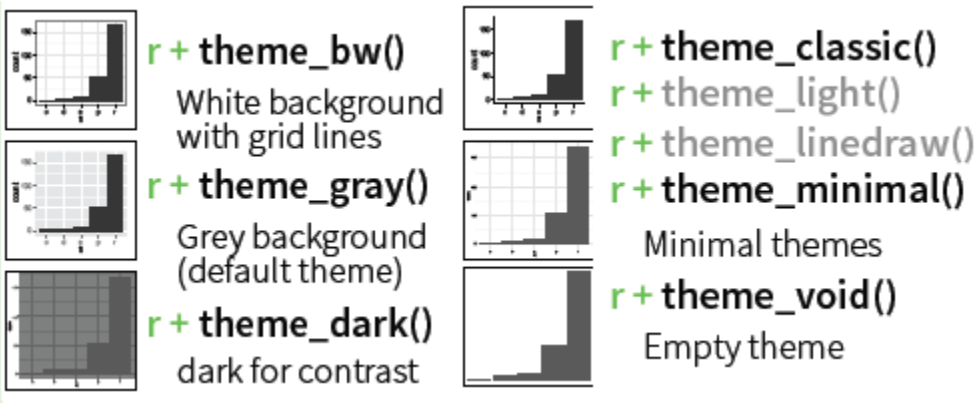

```
ggplot(mtcars, aes(x=hp, y=mpg, color=cyl, shape=cyl)) +
    geom_point(size=3) +
    geom_smooth(method="lm", aes(fill=cyl))
```

ggplot2: GoG -> graphic language

- The implementation of GoG ideas in ggplot2 for R created a more expressive language for data graphs
  - layers: graph elements combined with “+” (read: “and”)

```
ggplot(mtcars, aes(x=hp,  y=mpg)) +
    geom_point(aes(color = cyl)) +
    geom_smooth(method ="lm") +
```

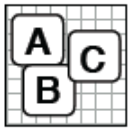
  - themes: change graphic elements consistently



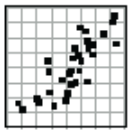
ggplot2: more geoms

Continuous X, Continuous Y

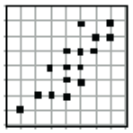
e <- ggplot(mpg, aes(cty, hwy))



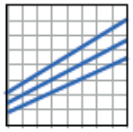
e + **geom_label**(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE)
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust



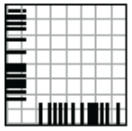
e + **geom_jitter**(height = 2, width = 2)
x, y, alpha, color, fill, shape, size



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

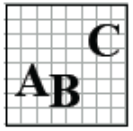


e + **geom_quantile**()
x, y, alpha, color, group, linetype, size, weight



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



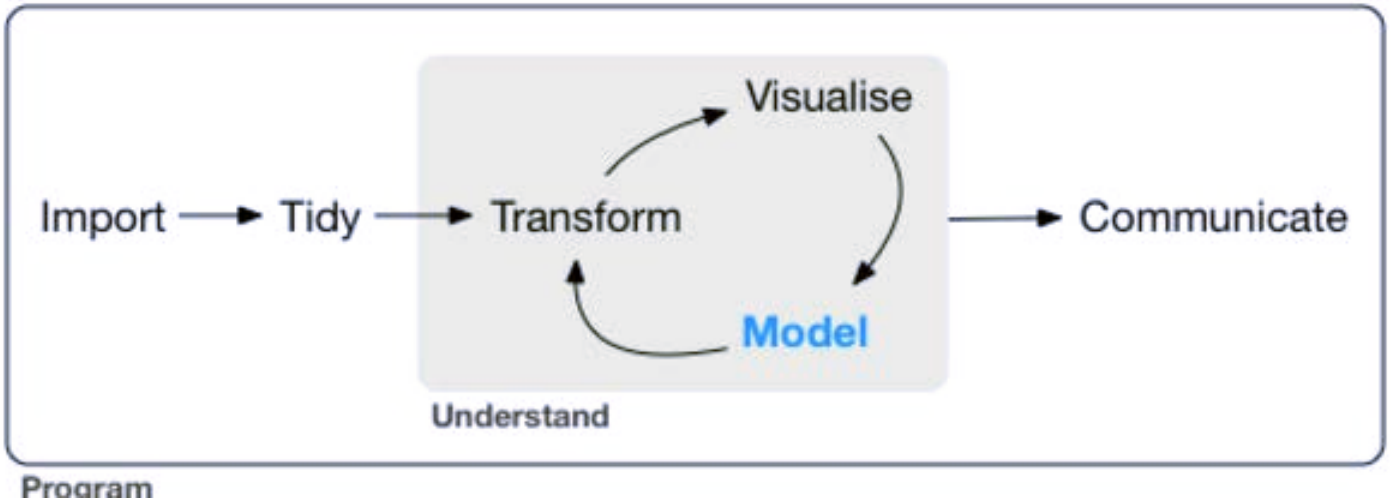
e + **geom_text**(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE)
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

ggplot2 facilitates graphical thinking by making a clear separation among:

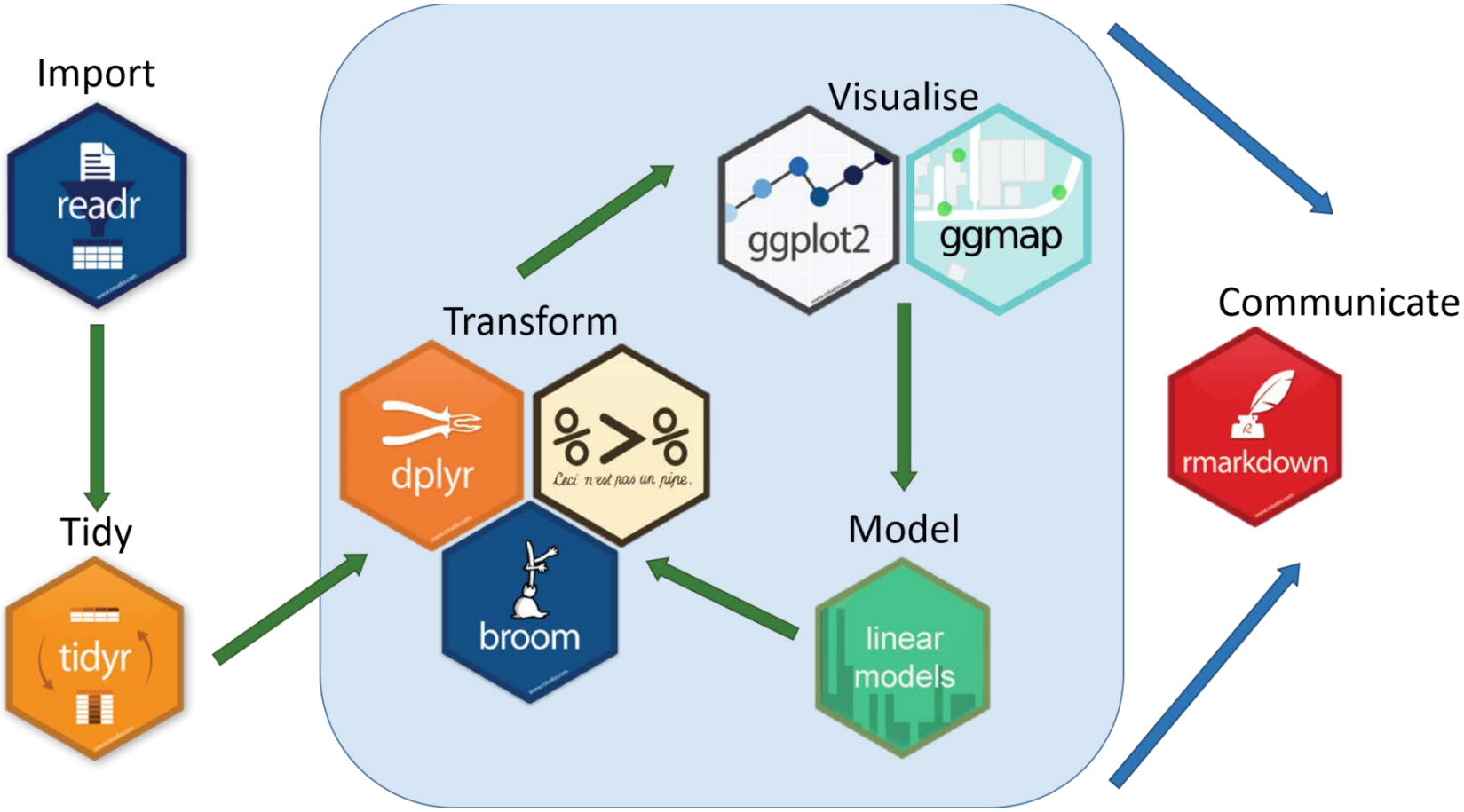
- mapping data variables to plot features (aes());
- geometric objects (geom_())
- statistical summaries (stat_())

A larger view: Data science

- Data science treats statistics & data visualization as parts of a larger process
  - Data import: text files, data bases, web scraping, …
  - Data cleaning → "tidy data"
  - Model building & visualization
  - Reproducible report writing



The tidyverse of R packages



Summary

- Graphical developers in the Golden Age recognized the idea of “graphic language,” but could not define it.
- Bertin first formalized the relations between graphical features (“retinal variables”), data attributes (O, Q, ≠, ≡), and “reading levels”
- Wilkinson, in GoG, created a comprehensive syntax and algebra to define any graph
- Wickham, in ggplot2, created an expressive language to ease the translation of graphic ideas into plots.