



GEOS 436 / 636

Programming and Automation for Geoscientists

– Week 07: Plotting –

Ronni Grapenthin
rgrapenthin@alaska.edu

Elvey 413B
x7682

How can you process your data
and
create publication-quality plots?

Philosophy

- Data change: daily updates, new field season, different analysis technique, forgot something ...
- Everything you do manually to a figure has to be repeated after a data update
- Automate figure creation: as much as possible have a figure generated without “photoshop interference”

HOW?

Set up your workflow such that ONE action initiates data ingestion, analysis, modeling, etc., and eventually figure creation.

Philosophy

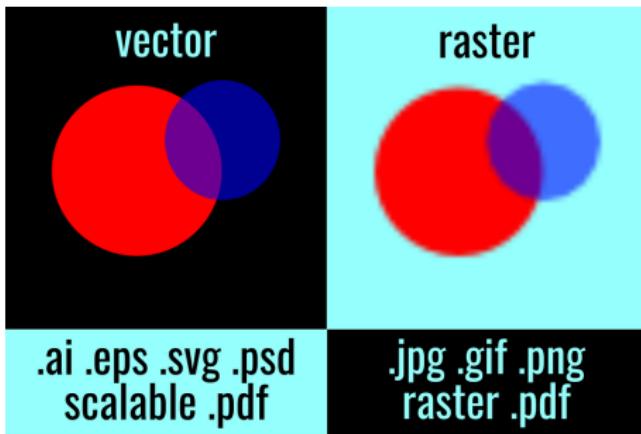
- Setting up a good figure can easily take a day or more – figures are major scientific products
- Good figures will get reused by others (hopefully crediting your work): advertisement of your work
- Know your audience:

lecture ≠ seminar ≠ journal article ≠ meeting presentation ≠ outreach

- Create figures in a way that makes them easily adaptable (vector graphics over raster graphics)

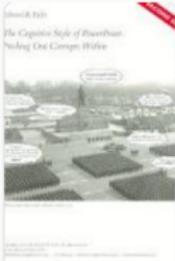
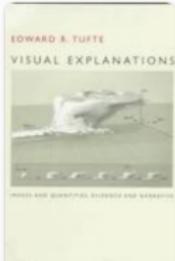
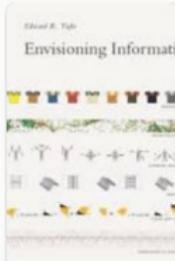
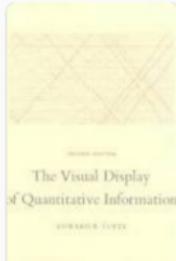
Philosophy

- Create figures in a way that makes them easily adaptable
- Use vector graphics over raster graphics where possible



Philosophy

- Check Edward Tufte books for some guidelines on what makes good plots.
- Most important advice: avoid “Chartjunk”

						
Beautiful Evidence 2006	The Cognitive Style of Pow... 2003	Visual Explanations 1997	Visual and Statistical T... 1997	Envisioning Information 1990	None: Paperback 1990	The Visual Display of Q... 1983

Chartjunk - Moirè Art

*"The interior decoration of graphics generates a lot of ink that does not tell the viewer anything new. The purpose of decoration varies – to make the graphic appear more scientific and precise, to enliven the display, to give the designer an opportunity to exercise artistic skills. Regardless of its cause, it is all **non-data-ink or redundant data-ink**, and it is often chartjunk."*

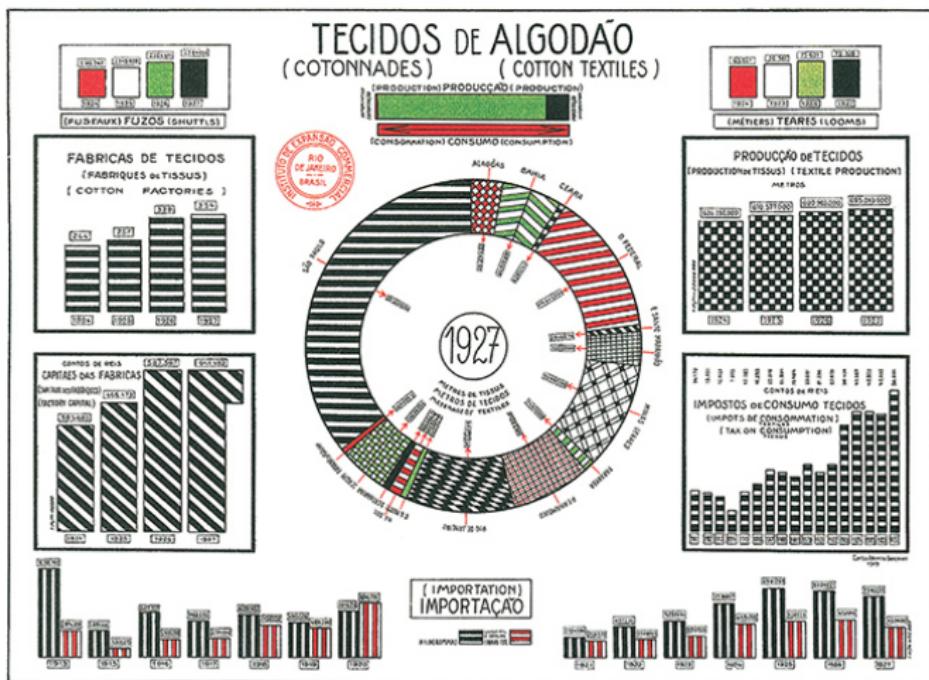
Tufte: The Visual Display of Quantitative Information

Carefully assess the purpose of the ink you're using.

Chartjunk - Moiré Art

This moiré vibration, probably the most common form of graphical clutter, is inevitably bad art and bad data graphics. The noise clouds the flow of information as these examples from technical and scientific publications illustrate:

Instituto de Expansão Commercial,
Brasil: *Graphicos Económicos-Estatísticas*
(Rio de Janeiro, 1929), 15.



Chartjunk - Moirè Art, Necker Effect

CHARTJUNK 109

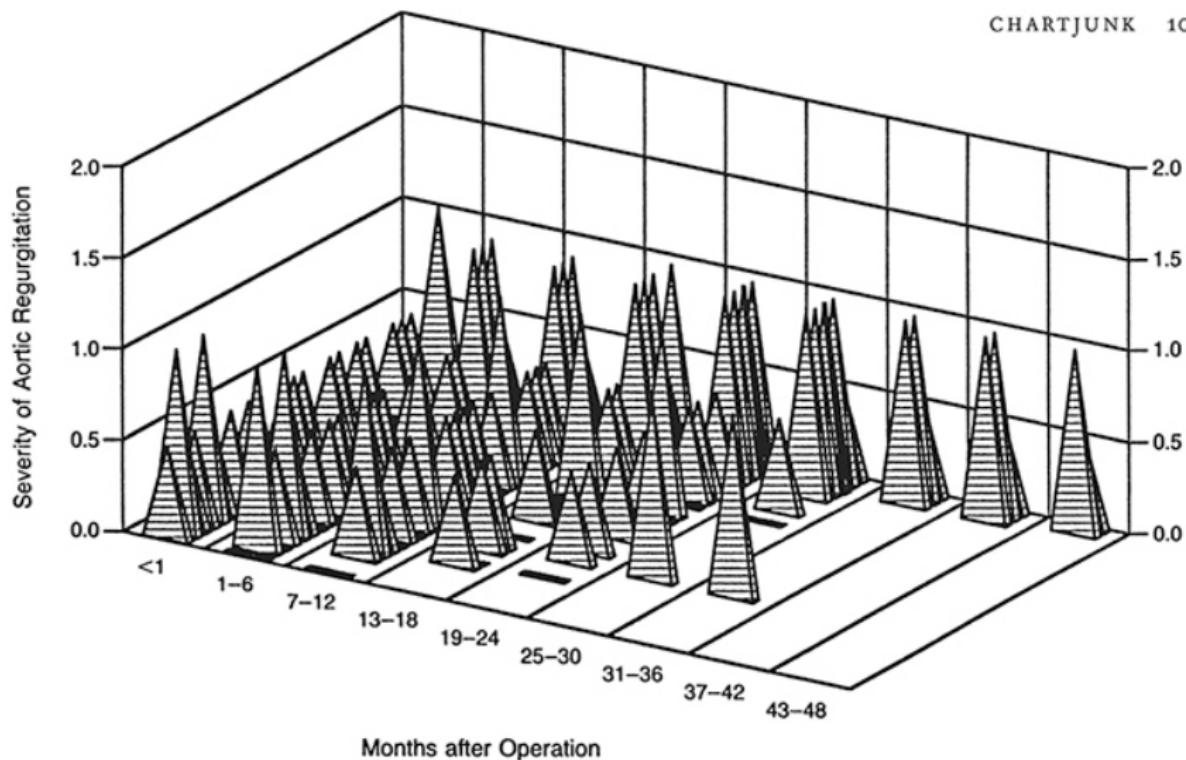
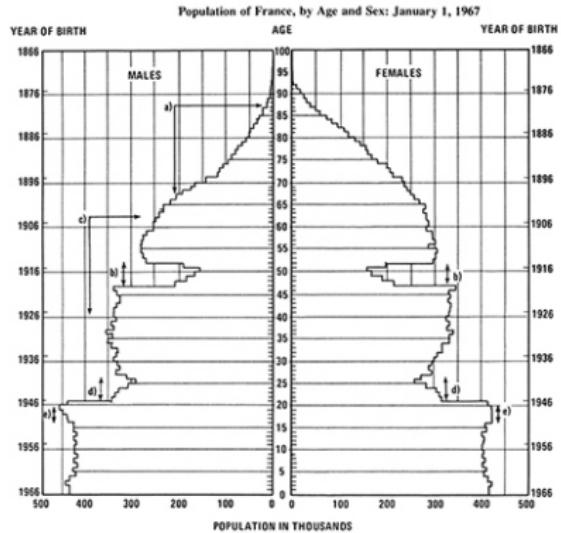


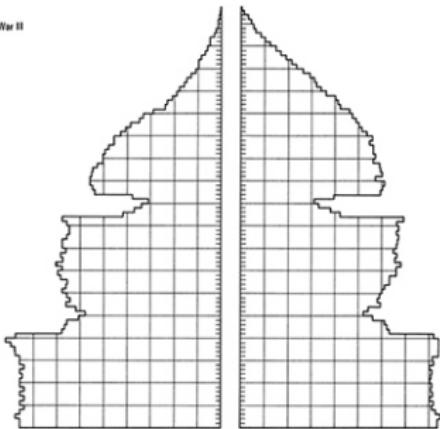
Figure 2. Serial Echocardiographic Assessments of the Severity of Regurgitation in 31 Patients. The numerical grades were assigned according to the severity of regurgitation, as follows: 0, none; 0.5, trivial; 1.0 to 1.5, mild; 2.0, moderate; and 3.0, severe.

Tufte: *The Visual Display of Quantitative Information*

Chartjunk - Gridlines



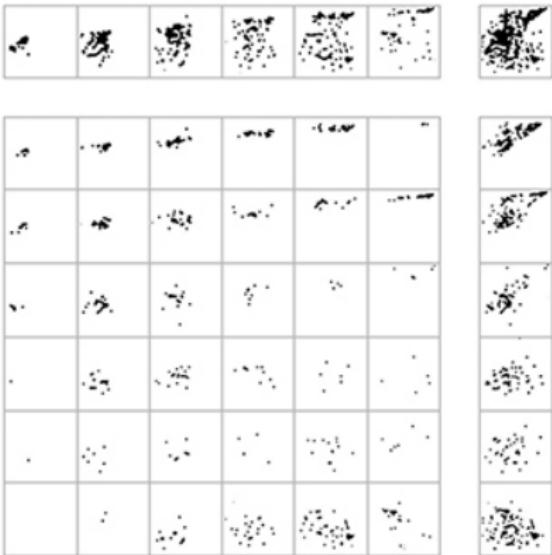
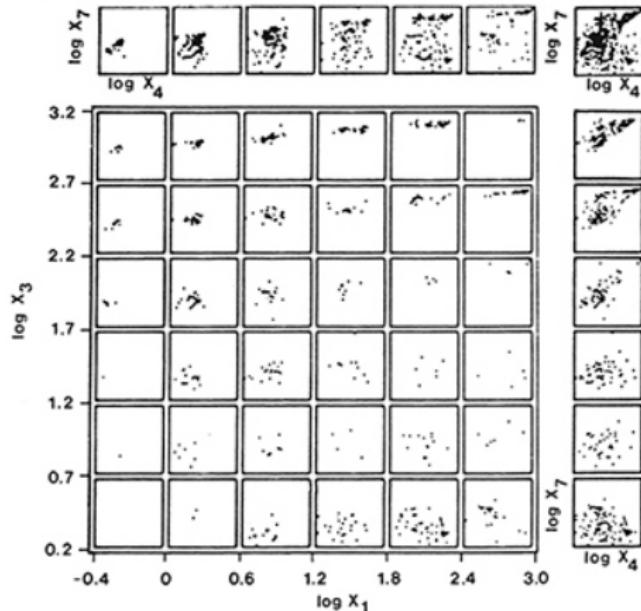
- (a) Military losses in World War I
- (b) Deficit of births during World War I
- (c) Military losses in World War II
- (d) Deficit of births during World War II
- (e) Rise of births due to demobilization after World War II



Tufte: *The Visual Display of Quantitative Information*

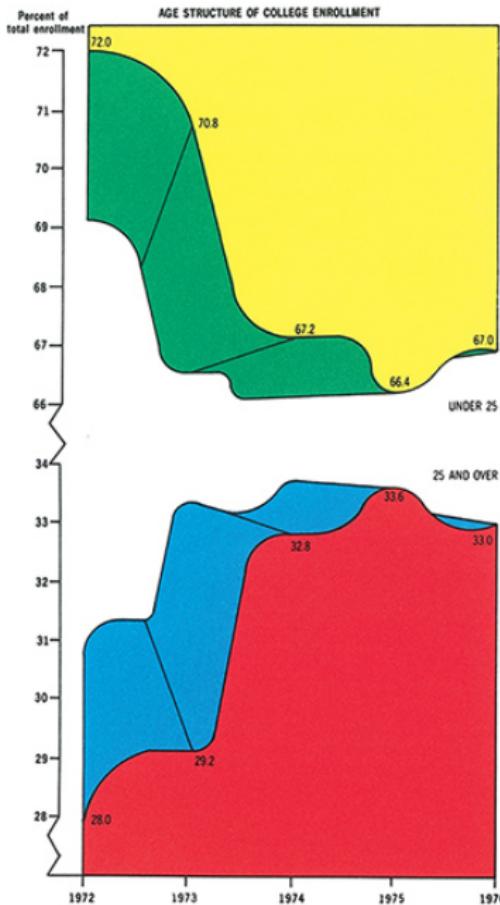
Chartjunk - Gridlines

MULTIWINDOW PLOT OF PARTICLE PHYSICS MOMENTUM DATA



Tufte: *The Visual Display of Quantitative Information*

Chartjunk - A Duck

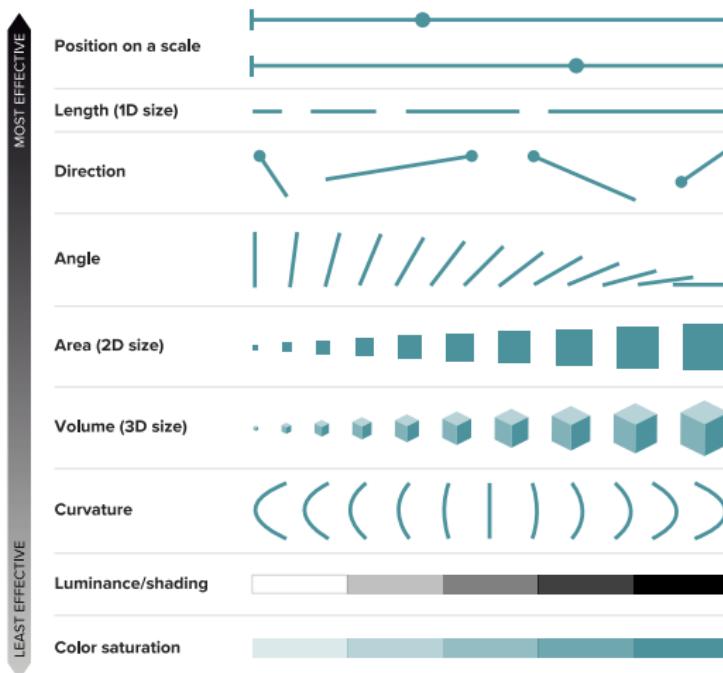


Tufte: *The Visual Display
of Quantitative Information*
(from "American Education"
Magazine, 1970s)

Visual Elements

Ranking of visual elements

Studies have identified the easiest ways for people to understand differences in quantitative data, on a scale from most effective to least.



Color Contrasts

Simultaneous contrast

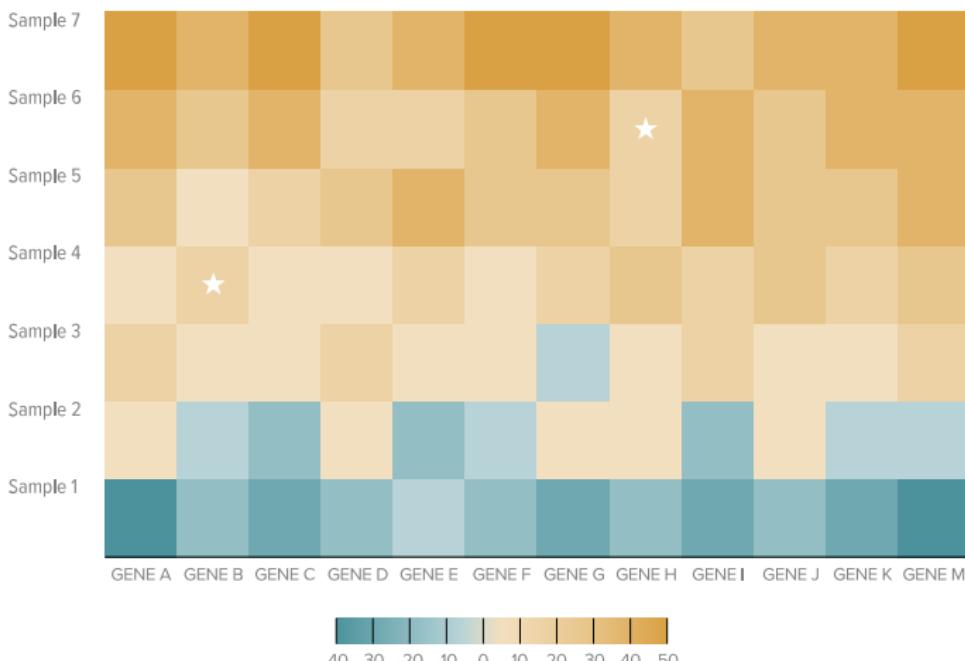


5W INFOGRAPHIC / KNOWABLE

Color Contrasts

Contrast can create illusions

Starred boxes are an identical shade of orange, despite their appearance.



Plotting Tools

- There are many tools.
- Don't get limited by the tools / defaults
- Invest some time to research which tool helps you realize the vision for your data visualization
- Program your own graphics if necessary (Python, GMT, 3djs, . . .)

Plotting Tools

Python:



Unix:



THE GENERIC MAPPING TOOLS



Matplotlib

pyplot API

`matplotlib.pyplot` is intended for simple cases of programmatic plot generation, it provides a set of functions that make Matplotlib work like MATLAB.

object-oriented API

If you need more control and customization, work directly with the figure objects that matplotlib generates

matplotlib.pyplot

matplotlib.pyplot

`matplotlib.pyplot` is a state-based interface to matplotlib. It provides a MATLAB-like way of plotting.

`pyplot` is mainly intended for interactive plots and simple cases of programmatic plot generation:

```
import numpy as np
import matplotlib.pyplot as plt

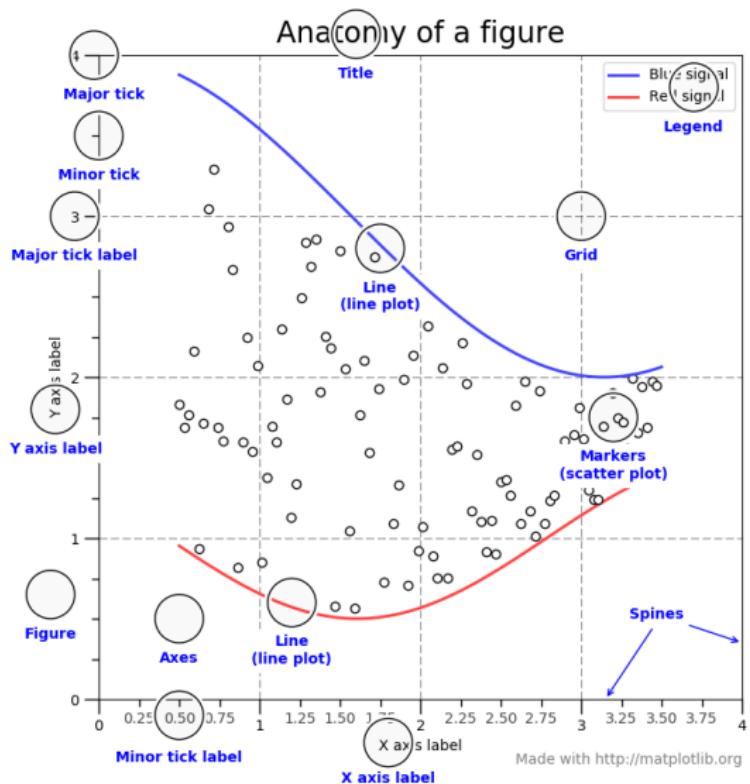
x = np.arange(0, 5, 0.1)
y = np.sin(x)
plt.plot(x, y)
```

The object-oriented API is recommended for more complex plots.

Functions

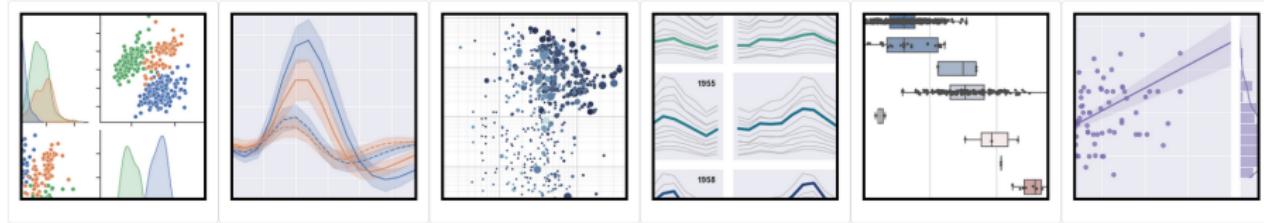
<code>acorr(x[, data])</code>	Plot the autocorrelation of <i>x</i> .
<code>angle_spectrum(x[, Fs, Fc, window, pad_to, ...])</code>	Plot the angle spectrum.
<code>annotate(text, xy, *args, **kwargs)</code>	Annotate the point <i>xy</i> with text <i>text</i> .
<code>arrow(x, y, dx, dy, *kwargs)</code>	Add an arrow to the axes.
<code>autoscale([enable, axis, tight])</code>	Autoscale the axis view to the data (toggle).
<code>autumn()</code>	Set the colormap to "autumn".
<code>axes([arg])</code>	Add an axes to the current figure and make it the current axes.
<code>axhline([y, xmin, xmax])</code>	Add a horizontal line across the axis.
<code>axhspan(ymin, ymax[, xmin, xmax])</code>	Add a horizontal span (rectangle) across the axis.
<code>axis(*args[, emit])</code>	Convenience method to get or set some axis properties.
<code>axline(xy1[, xy2, slope])</code>	Add an infinitely long straight line.
<code>axvline([x, ymin, ymax])</code>	Add a vertical line across the axes.
<code>axvspan(xmin, xmax[, ymin, ymax])</code>	Add a vertical span (rectangle) across the axes.
<code>bar(x, height[, width, bottom, align, data])</code>	Make a bar plot.
<code>barbs(*args[, data])</code>	Plot a 2D field of barbs.
<code>barh(y, width[, height, left, align])</code>	Make a horizontal bar plot.
<code>bone()</code>	Set the colormap to "bone".
<code>box([on])</code>	Turn the axes box on or off on the current axes.
<code>boxplot(x[, notch, sym, vert, whis, ...])</code>	Make a box and whisker plot.
<code>broken_barh(xranges, yrangle, *, data)</code>	Plot a horizontal sequence of rectangles.
<code>cla()</code>	Clear the current axes.

Matplotlib Figure Anatomy



Seaborn

seaborn: statistical data visualization



seaborn.pydata.org

“Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.”