

# Visualization

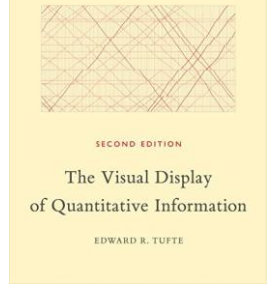
# Types of visualizations

- Data/information visualization (“charts”)
- Scientific visualization
- Scientific illustrations / cartoons (can use in combination with programming but not necessarily)
  
- Static
- Dynamic
  - animation – GIF, movies
  - graphical user interface (GUI) / interactive plots

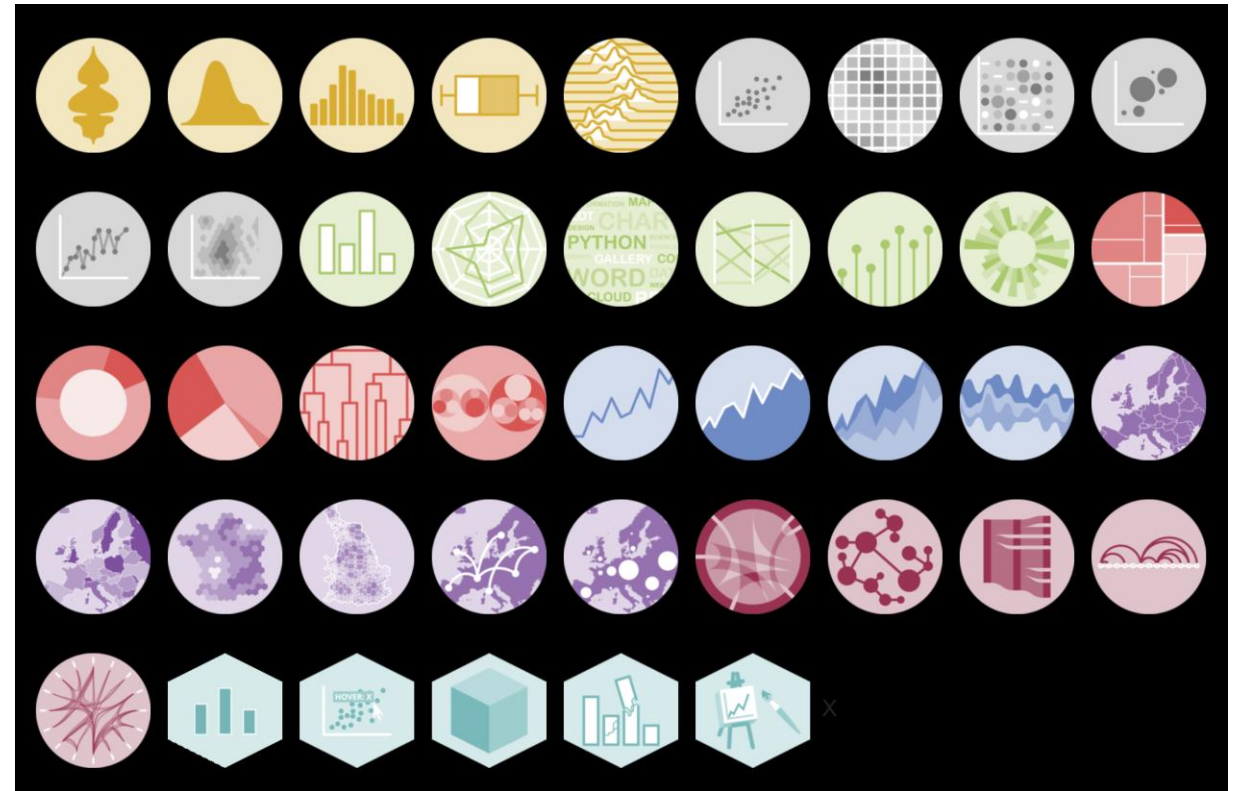
# When to use which?

- Who is the audience?
  - Internal (project team) / external (client)
- How much time does it take the audience to digest the information?
  - A sequence of snapshots may be preferred over an animation or movie
- Is an interactive plot necessary?
  - if the parameter dimensionality/space to explore is large and nonlinear
  - does the client need an interactive plotting tool or can/should *you* summarize the results
- Do aesthetic enhancements obfuscate the important trends?

# Charts

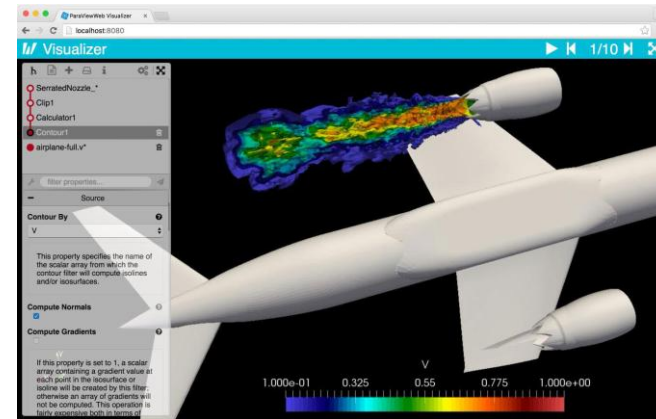
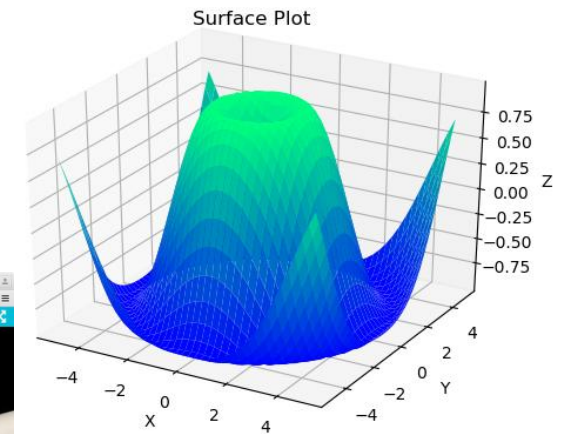
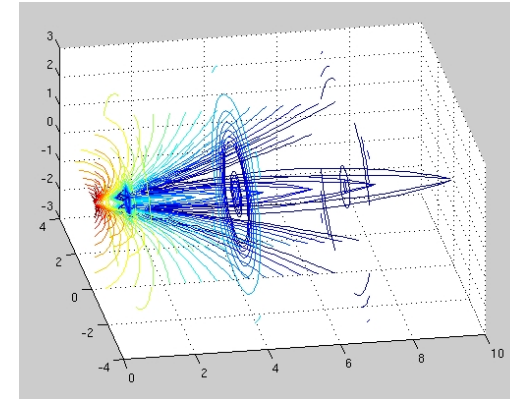


- Summarize data
- Can be generated programmatically in many data analysis languages
- Common method of visualization
- Use only as much “ink” as necessary

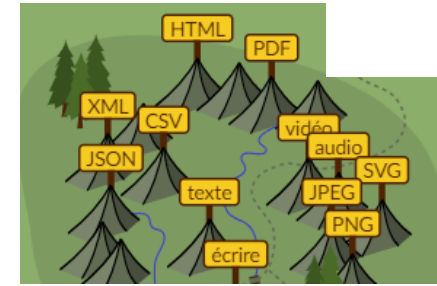


# Scientific visualization

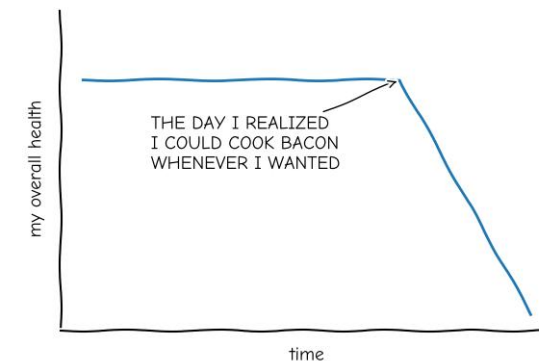
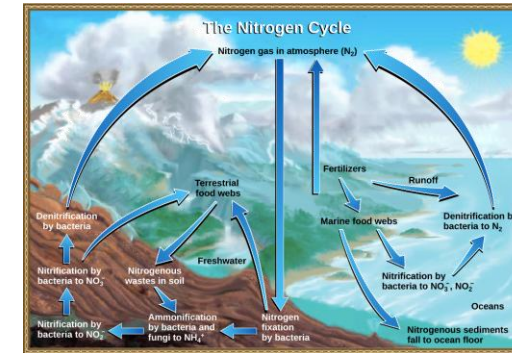
- Visualize (vector) fields
- Often (but not always) dealing with physical representations
- May involve rendering (adding realistic reflections, shading, etc.)
- Can do some in Python/MATLAB
- Often requires specialized tools
  - OpenGL
  - VTK
  - MayaVi
  - Blender
  - Rhinoceros 3D



# Scientific illustrations / cartoons



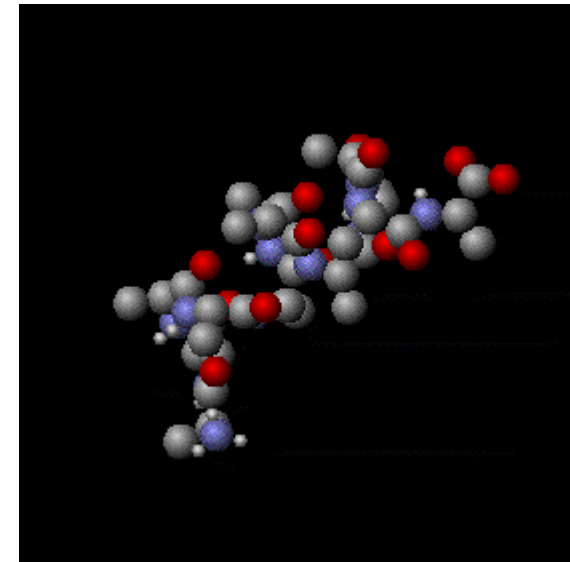
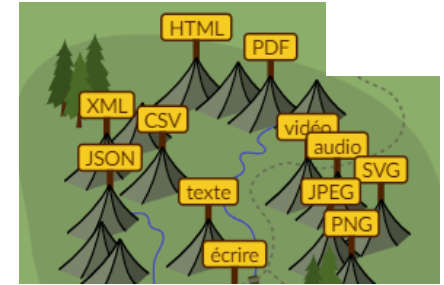
- By hand (traditional medium)
- **Raster graphics** (jpg, gif, png)
  - Adobe Photoshop / GIMP
  - efficient storage for images
- **Vector graphics** (pdf, svg)
  - Adobe Illustrator / Inkscape
  - small files for simple plots (points and lines), large files for complex plots (many points or images)
- Cartoons – useful for showing imprecise nature of the illustrated data



"Stove Ownership" from xkcd by Randall Munroe

# Animation

- GIFs: sequence of images
  - runs on any machine
- Movies: sequence of images stored as “frames”, optional accompanying audio track
  - more compact than GIFs
  - compressed with codec – may require OS/platform-specific codec to decode – need to test on different machines
- Can make with Python, MATLAB, and other tools
- Good for presentations, but snapshots of animation is more suitable for static technical reports



# User interfaces

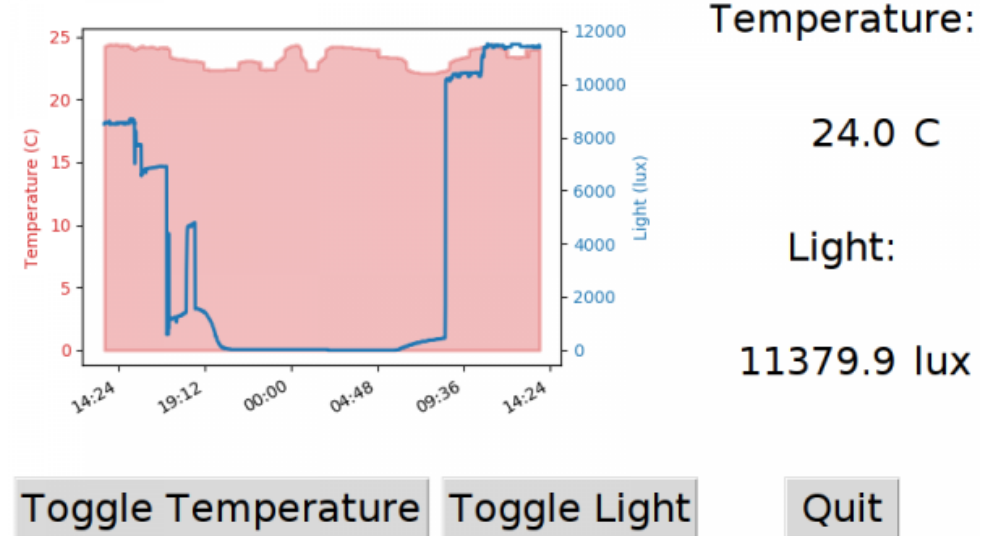
- Command line interface (CLI)
  - run program (with input files)
  - run program with input arguments
  - prompt user input from keyboard
- Graphical user interface (GUI)
  - toolkits – e.g., Tkinter, wxPython, PyQt
  - dashboards (web application) – e.g., Dash, Streamlit, Shiny
  - web application on web framework – e.g., Flask/Django with client-side (front end)
  - app – e.g., Dart

## Use cases

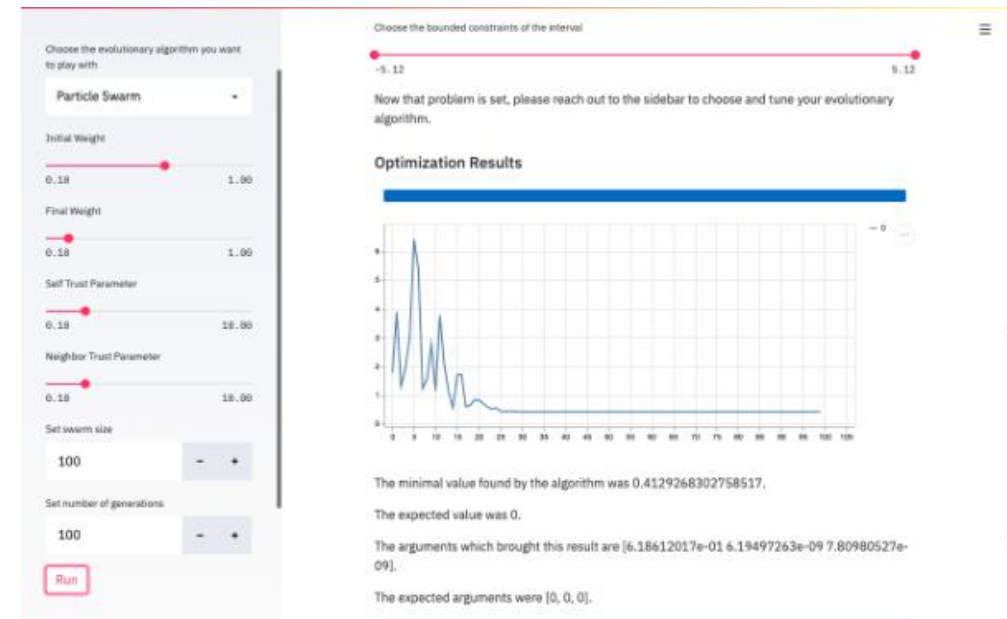
- CLI
  - for other technical users
  - programmable and reproducible (when prompting for user input from the keyboard, input must be saved to be reproducible)
- GUIs
  - for technical and semitechnical users
  - the question to be answered by your program is not completely predefined
  - explore high-dimensional spaces

# Graphical User Interfaces (GUIs)

- Drive program execution through graphical elements rather than command line arguments or file input parameters
- Uses – “Human Intelligence Tasks”
  - interactive plots – high-dimensional data exploration
  - data labeling
- Implementation
  - desktop/mobile app
  - web app (“dashboards”)
- Some GUI elements are built into interactive plots (MATLAB, matplotlib, etc.)
  - scatterplot brushing
  - spanning



<https://learn.sparkfun.com/tutorials/python-gui-guide-introduction-to-tkinter/all>



<https://streamlit.io/>

**Not part of ENG-270 project!**

# Do you need a user interface?

## **GUIs**

- most stakeholders want environmental engineers to propose the solution – not provide a tool for them to use (even if eye candy)
- if a GUI is needed, environmental engineers provide the computational engine; GUI development is outsourced to a specialist
- in some cases, environmental engineer will have to develop a GUI when 1) the team is small, 2) the engineer has intimate knowledge of the data and how best it should be represented

## **Preferred alternatives**

- For each scenario, save input parameters to file – e.g., JSON, CSV
- Code loops through each scenario and generates outputs saved to different files / folders
- Another code harvests results and constructs summary tables and plots

# Figures in reports

- Make axis labels "comically large." (*common error* – hard-to-read axes)
- Typically don't need plot titles at top unless multi-panel.
- The **main text** should include interpretation of the figure. The **caption** includes description of the elements of the figure.
- Often it is unnecessary to say, "Figure X shows {statement of finding}." Say instead: "{Statement of finding} (Figure X)" – unless the figure contains multiple findings that should be expounded upon in detail.