Topic 12: Plotting & Data Visualization
W200 Python for Data Science

Slide 1:

Introduction: the purpose of plotting & data vis for this course is primarily to help you think about your data, how you can integrate tools to represent data in a visual language, and so facilitate presenting, explaining, and exploring the data.

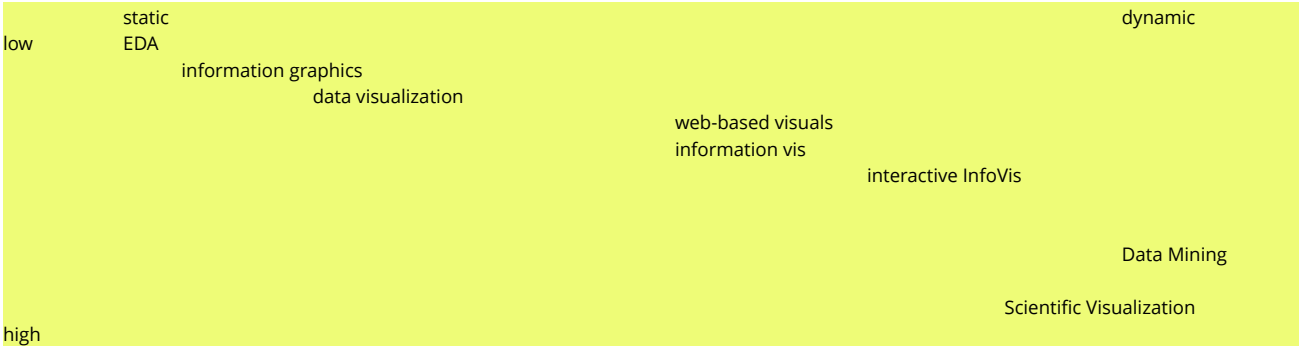
Slide 2:

"Visualizing" data started in the 1940s, when *American Demographics* journal's lead article was a picture of the data, instead of the end-less columns of numeric values. From there the work languishes 'til the rise of Data- and Text-Mining. The thought is that originally converting data from SQL structured to full-text exploration of patterns required both a statistician to distinguish between types of "interesting events" (that is, patterns that seem to be statistically sound) and a domain expert, to ensure that what was happening was even feasible given the subject matter. The huge amounts of data and the complexities of visualizing links between the data - let alone the amount of computing power necessary - meant visualization was part of big companies' efforts - SAS, SPSS, IBM and the like.

Today because computers are used to create graphic images easily, accounting software providing easy-to-create plots, and because of the Internet, imho, everyone thinks that visualizing data is (a) useful, (b) necessary, (c) and naturally appropriate and (d) comprehensible. This isn't true.

In fact we ought to think of data vis primarily as "EDA" - exploratory data analysis and plotting. EDA means bar charts, line graphs, etc., - everything you see in MS Excel or Apple Numbers. It is merely a static representation of a given state-of-affairs of the data.

There is a continuum of terms and work areas related to visualization, depending on the level of dynamism in the data and the amount of computation:



Definition: Let's define data to be primarily the visual expression of data using graphs, plots, usually static … and considering the graphic or aesthetic qualities that help to present, explain, and explore data.

What might be looking for? First the data should be as neutral a presentation as possible. We can by accident or design combine data or model and display the data such that we swing an interpretation.

Second the data need to be applicable to the topic and to the end-users' needs - hence able to be interpreted by them as appropriate, understandable, applicable, and so enable them to take action. This is the key - to be "informed" - as opposed to a passive acknowledgement of some bits of data.

Third, ask yourself WHY you're creating this viz? To be pretty? To satisfy the boss? To fill up a report?

Slide 3:

DISCUSSION

there's no right/wrong answers - just express your thought about the topic to get your thinking ...

Slide 4:

There are many code libraries to help you express your data. Why so many? Some are optimized for different purposes, such as ggplot, that works closely with python and the stats software R. Plotly is not OpenSource, Bokey is ... both are for increased interactivity with the end-user. Interactivity equates to mouse events.

Don't forget that plotting data can integrate dots, lines, squares, circles, etc. Collectively these are "visual primitives". But we can include, too, real pictures, realistic maps, different symbol sets ... and manipulate the size, placement, color, etc., of the whole to engage the end-user in a kind of discussion with the data.

Slide 5:

The first library to know is matplotlib. Just as NumPy and Pandas provides abilities to Python, so matplotlib provide better visualization techniques to Python than python's own built-in graphic set.

Most of the visualization libraries depend on matplotlib. One of these very useful tools is Seaborn.

How to know which to use? First is the purpose of the visualizations. Second is your own knowledge-base and/or the company's. Kind of useless to write in a tool you know if none of your colleagues does. There's also the issue of maintenance and scalability of your projects.

Another reason is the syntax. Libraries of code have their own syntax ... and trying to learn a dozen of them is not really useful.

Breakout Activity 2 [Week_12_Activity_(Plotting).ipynb]

Slide 6:

How to approach creating a visualization? There are some usual steps we take. This diagram may be a useful model.

- Know your client and your data
- Data may come from a variety of resources and at various rates of ingesting

- Inspect your data - before you can use them for machine learning and analysis, you need to know the domain and range of the data, what you're cleaning for, etc.
- Once you've inspected the data it's an iterative process for reducing (classification and clustering) of the data, determining new data from the existing set …
- What data are you going to use? Check the data match the need … and get to know more about visual languages and graphic design in order to know what *visual model* supports the *data model*.
- Once structured, there is a whole world of work (information visualization) with lots of theories … in general we might what to look at the "expressiveness" of the data - what they imply, user issues, approachability of the visualization (meaning will the person want to engage with it and how long, etc.)
  - Effectiveness is not unlike common measures of time/task completion or time-on-task measures [common HCI measures].
- Finally, you'll want to think adjust for the optimal user experience.

Slide 7:

As you know from the asynch part, we can build piece-by-piece a visualization and, as usual, then find ways to make our building more efficient. One way might be to start to divide your work: modularize it so you can keep the *presentation* of the data separate from the *data themselves*. All term I've mentioned how this is a fundamental practice today. So …

- Data: what data structures are best? What analysis and processing tools are appropriate?
- Presentation of the data: What visual model suits your work? For example would you want a polar graph if the story of the data you're telling is linear, like a time line?
- Presentation of the data: what visual language elements will you use? What is the role of color and of fonts? (Graphic design) …

In the end do all cohere to make a "truthful" representation of the data?

Slide 8:

How can we build step-by-step the *code* behind the visuals?

(read and discuss each line)

Slide 9:

The Big Picture - getting an overview of rich and glorious world of visualizations!

This example is from d3.js, a JavaScript library used in W206.

Slide 10:

This chart is extracted from Börner's *Atlas of knowledge*. We will discuss a very few of the very many techniques out there.

Slide 11:

Since there are so many, here are few that you'll probably want to focus on …

[read and discuss briefly; show online images as needed]

Slide 12:

Return now to the code aspect of how we think about building …
import the libraries and think about the output device (Notebooks? Other?)
define the parameters of the data

determine the type of graph (here, for instance, a line *plot* … and this kind of graph requires data on the x and y axes and a color parameter. Here we see the shortcut "r" for red.

Provide a title to help people understand the data …

Pass the values and plots to the box holding the data (the "figure" in this case)

Slide 13:

Resources: Students' interest and backgrounds in visualization range a lot. So, we've posted some resources to help you for your projects and for your own study.
[demo]

Slide 14:

Resources: There are also countless monographs on the topic. I divide these into a couple of groups.

Communicative approaches
One is my own approach - an ethical, aesthetic, and technical combination that focuses on the end-user's "communication" with the data.

Empirical approach
Ware, Stanko, Maes and many others from the hard-core CS world view visualizations as one of the empirical approaches, meaning casting the viewer as a black-box of input and output of experience to be measured. Usually these studies start with the biology of sight and quantification.

Tufte - a statistician by training and an art historian by preference. His series of work demonstrate the history of efforts to express numbers in a variety of visual, tactile, inferential ways … as well as how we approach collapsing time, space, learning, and social/political into a 2d printed world.

Steele & Illinsky are popular but rather dictatorial leaders. They have their way and your way is wrong. [Just kidding but not far from an accurate statement.]

Börner and most of the research literature discuss "one-offs" … rather "this is how we did it." I don't see much of a building chain here. Jeff Herr, on the other hand, is in this realm but his world includes what *happens in digital worlds* when the motivation is the visual. What do people do and how do they respond. These questions are definitely not computer science ones … or are they?

Munzner, Mereilles, Castro, Chan and many others try to bridge the tech and art world. They vary in degree of tech know-how, visual know-how ... Almost none are actually graphic designers. Northeastern Univ, NYU, and other schools have started "MFA in Data Visualization" programs.

Slide 15:

How do you and your clients keep up to date? Reading professional and popular press journals/ websites is one way to go. Here we drift into Business Intelligence (BI), technology reviews, and return to white papers. Tableau, SAS, others, have papers about using [their products] visualization on your job.

Tableau - recommend you experiment with it. As a student you get a free license (I believe); an instructor for a class definitely can for a full year for your studies.

More? Feel free to contact us for resources, product demonstrations, and examples of actual practice.

Breakout Activity 2 [12.8-Drill-HeartDisease.ipynb]

file name: Week12-Plotting-NotesForInstructors.rtf