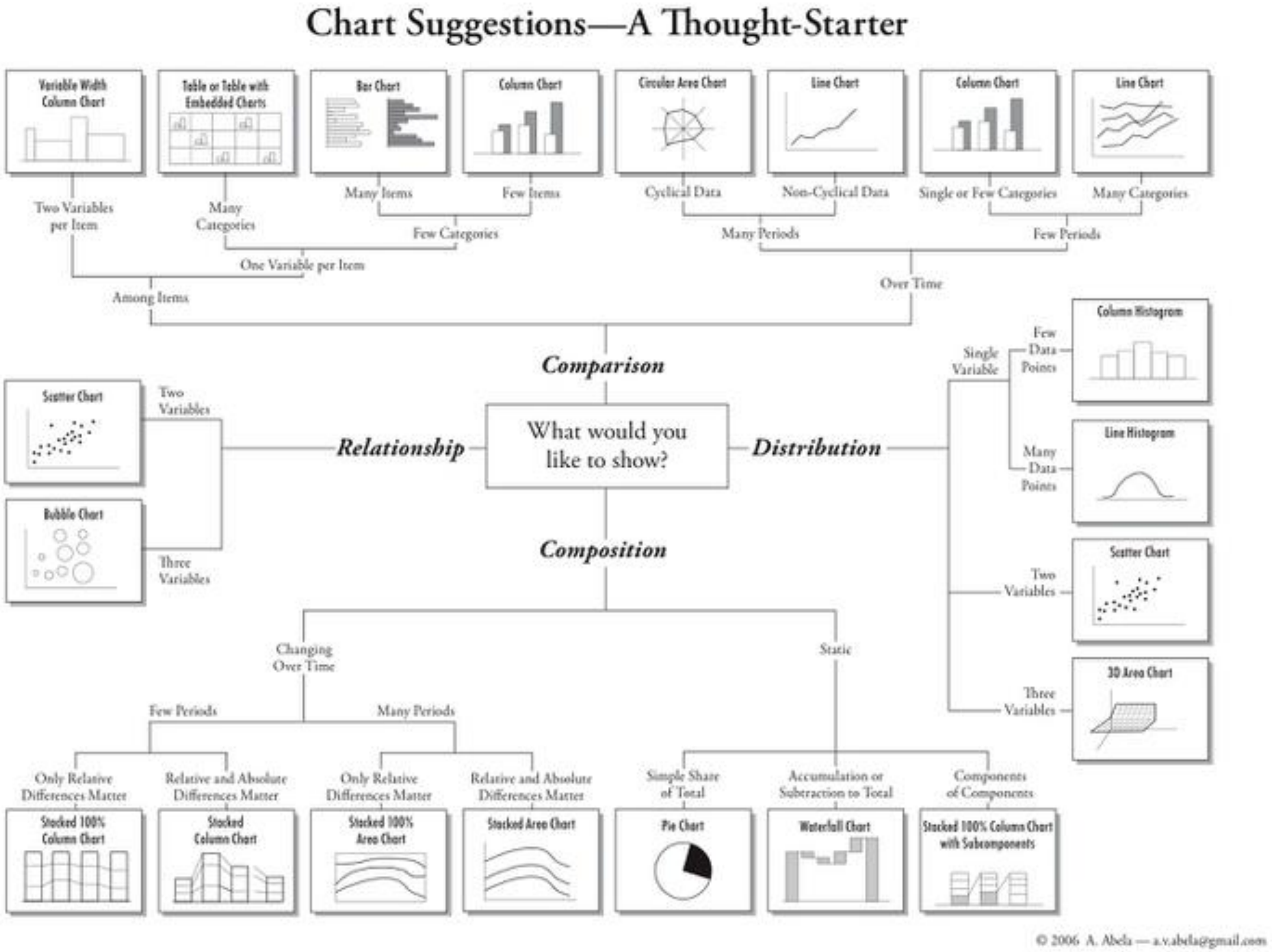
# COMP9321:
# Data services engineering

## Term1, 2021

Week 4: Data Visualisation

# The right paradigm



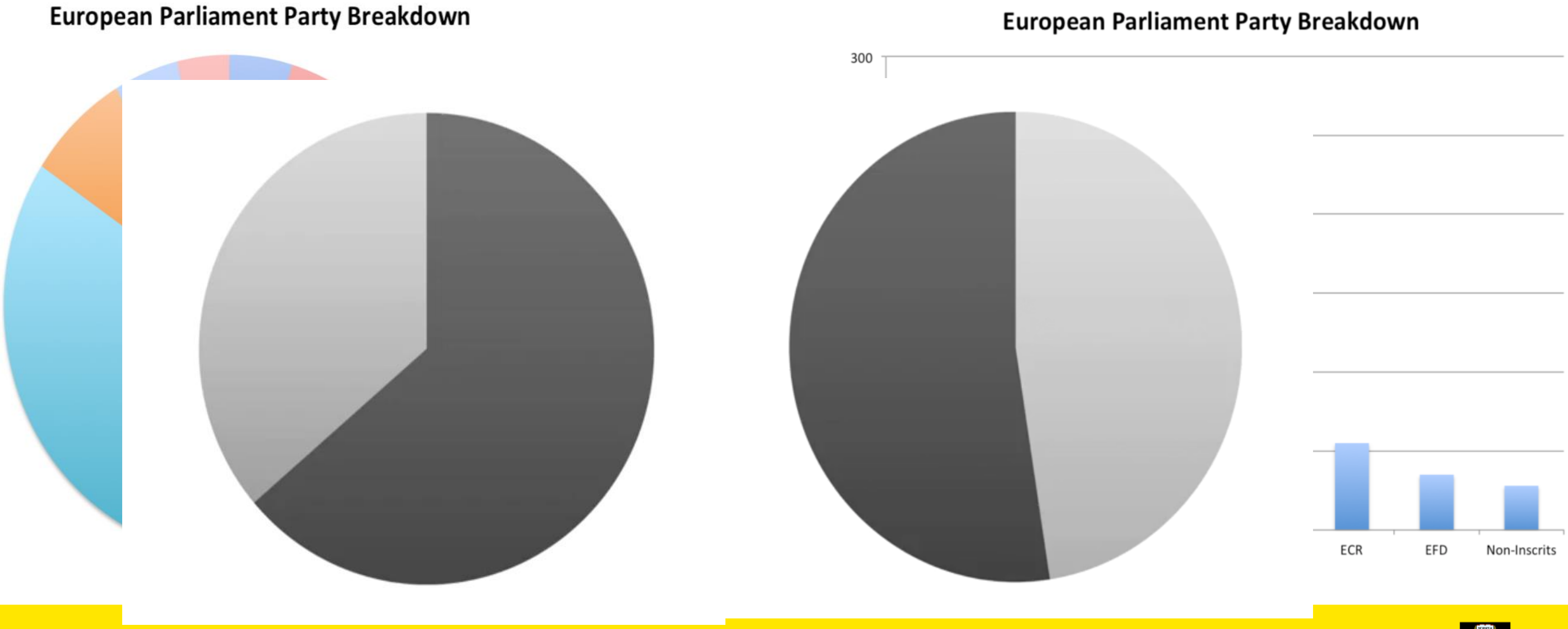
© 2006 A. Abela — a.v.abela@gmail.com

http://extremepresentation.typepad.com/files/choosing-a-good-chart-09.pdf

# The right paradigm

Pie Charts:

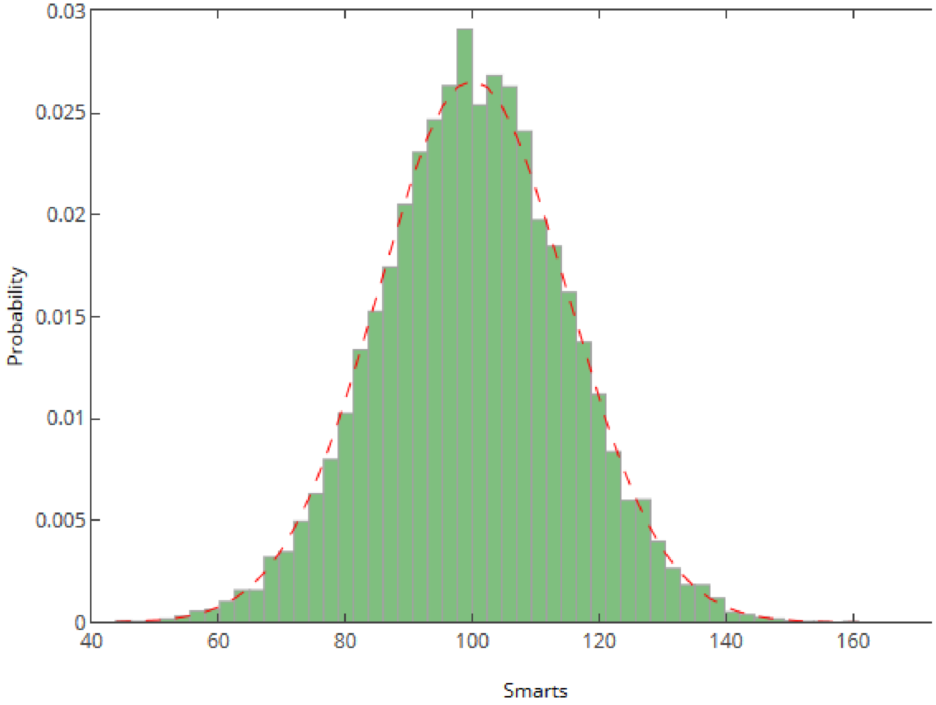
- Although commonly used, not considered an effective form. Human brain is not wired to parse round shape areas and arcs
- Normally other graphs can do the same job (e.g., BAR graphs)
- Maybe OK when showing two variables (<, >, similar, etc.)



# The right paradigm

## Histograms

- Histograms are useful for viewing (or really discovering)the distribution of data points
- The use of bins (discretization) really helps us see the “bigger picture” where as if we use all of the data points without discrete bins, there would probably be a lot of noise in the visualization, making it hard to see what is really going on



# The right paradigm: hierarchical data

To show the “connections” between and the hierarchy of objects.

A tree diagram:

http://mbostock.github.io/d3/talk/20111018/tree.html

- Default for showing hierarchy (e.g., org chart)
- Any situation where you a parent, which has children (and grand children)

A node link diagram:

http://mbostock.github.io/d3/talk/20111116/force-collapsible.html

- Showing a lot of links between objects

Tree map:

http://mbostock.github.io/d3/talk/20111018/treemap.html

- Size of each category

Chord Diagram:

https://bost.ocks.org/mike/uberdata/

Complex data -> very difficult to parse the information (e.g., between dots far apart)
Interactivity could help parse
(https://bost.ocks.org/mike/fisheye/)

# The right paradigm: showing data on maps

On an existing map API like Google API …

Place markers (https://www.latlong.net)

- specific location (e.g., building), centre of a region

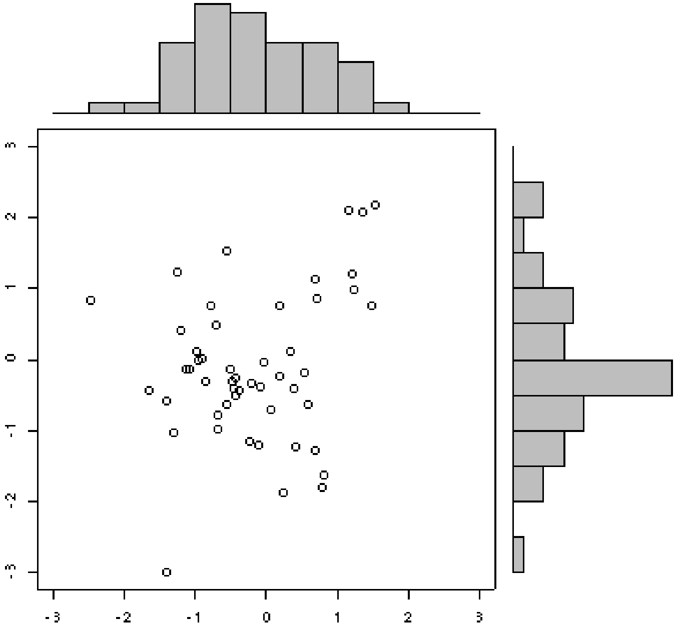
Layers (data associated with the regions on a map)

- Point clustering ( http://bl.ocks.org/andrewxhill/raw/8360694/)
  - display aggregated number/data points per region
- Choropleth map (http://leafletjs.com/examples/choropleth/)
  - display divided geographical areas or regions that are coloured, shaded or patterned in relation to a data variable.
- Heat map (https://onemilliontweetmap.com/)
- Flow map
  - show the movement of information or objects from one location to another and their amount (thickness of lines, colours)
  - https://datavizcatalogue.com/methods/flow_map.html
  - https://www.iom.int/world-migration

# Three tricks for doing more with less

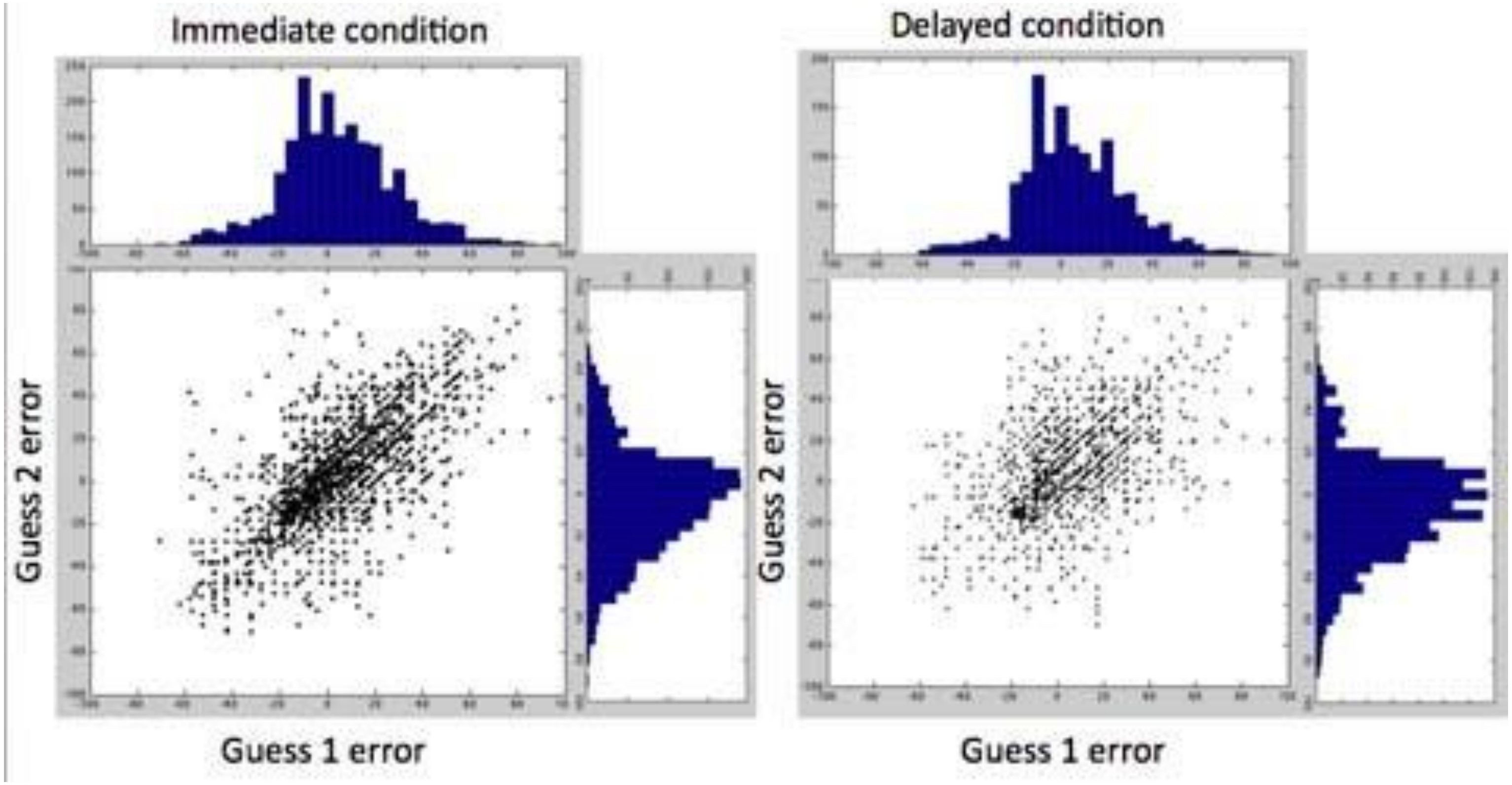
- Multiple plots
  - simple, easily interpretable subplots
  - can be beautiful but overwhelming
- Hybrid plots
  - a scatter plot of histograms
  - or a venn-diagram of histograms, etc.
- Multiple axes
  - plot two (or more) different things on one graph

# Hybrid plots

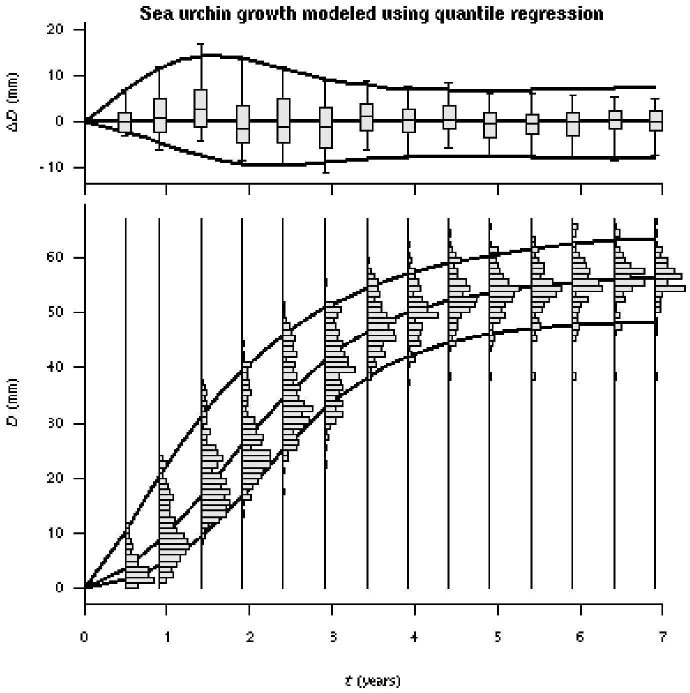


Courtesy of http://addictedtor.free.fr/graphiques/addNote.php?graph=78

# Hybrid plots



# Hybrid plots



Courtesy of http://addictedtor.free.fr/graphiques/addNote.php?graph=109

# Multiple plots

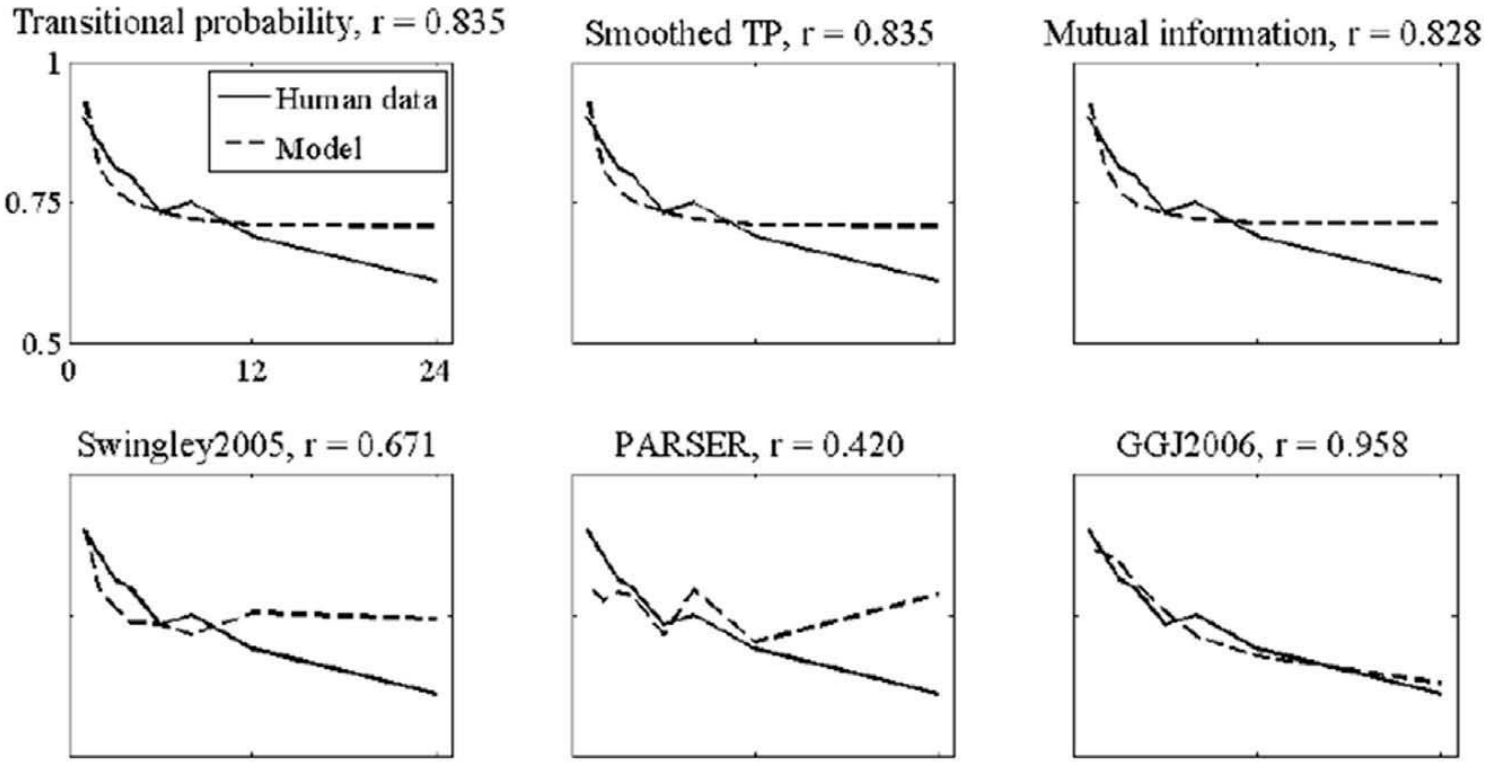


Figure 2. Best linear fit of each model's performance to human data, graphed by sentence length. The vertical axis represents decision probabilities for models and percentage correct for human data; the horizontal, sentence length.

Courtesy of Cognitive Science Society. Used with permission.

# Multiple plots

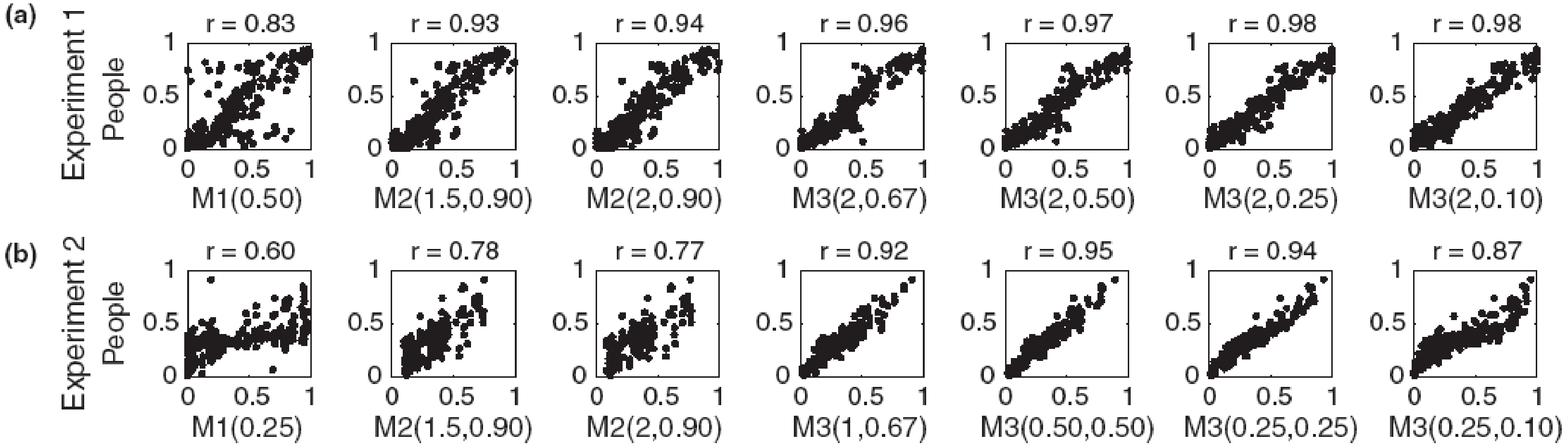
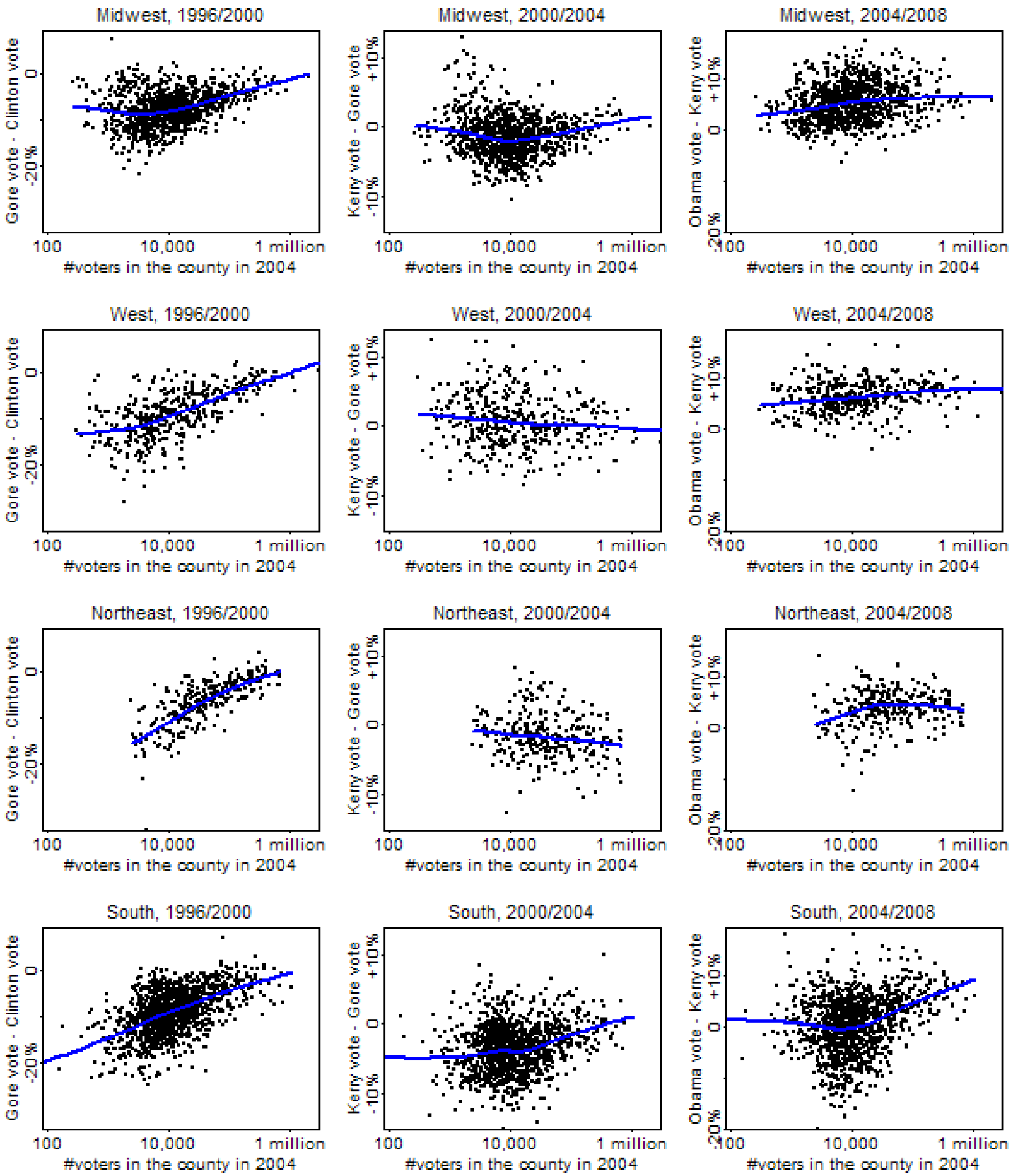


Figure 3: Example scatter plots of model predictions against subject ratings. Plots of model predictions use the parameter settings with the highest correlation from each model column of Tables 1 and 2. (a) Experiment 1 results. (b) Experiment 2 results.

Courtesy of Cognitive Science Society. Used with permission.

Baker, Tenenbaum, & Saxe (2007)

# Multiple plots



Courtesy of Andrew Gelman. Used with permission.

# Multiple axes

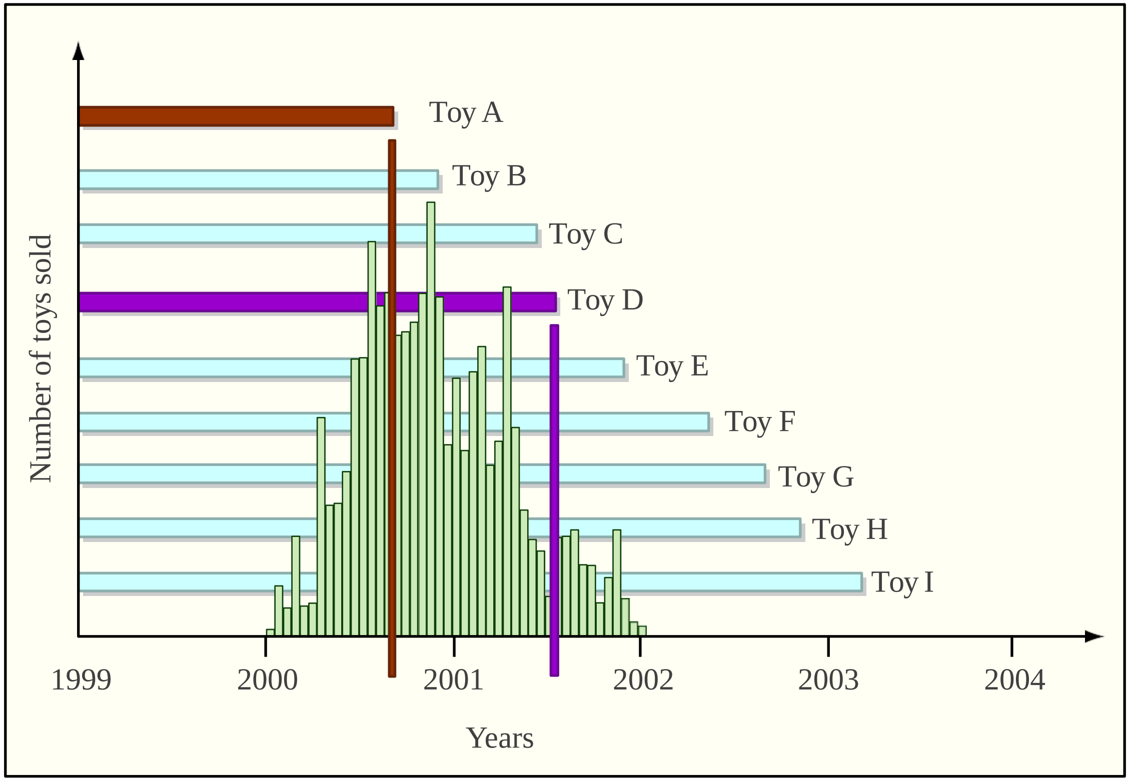
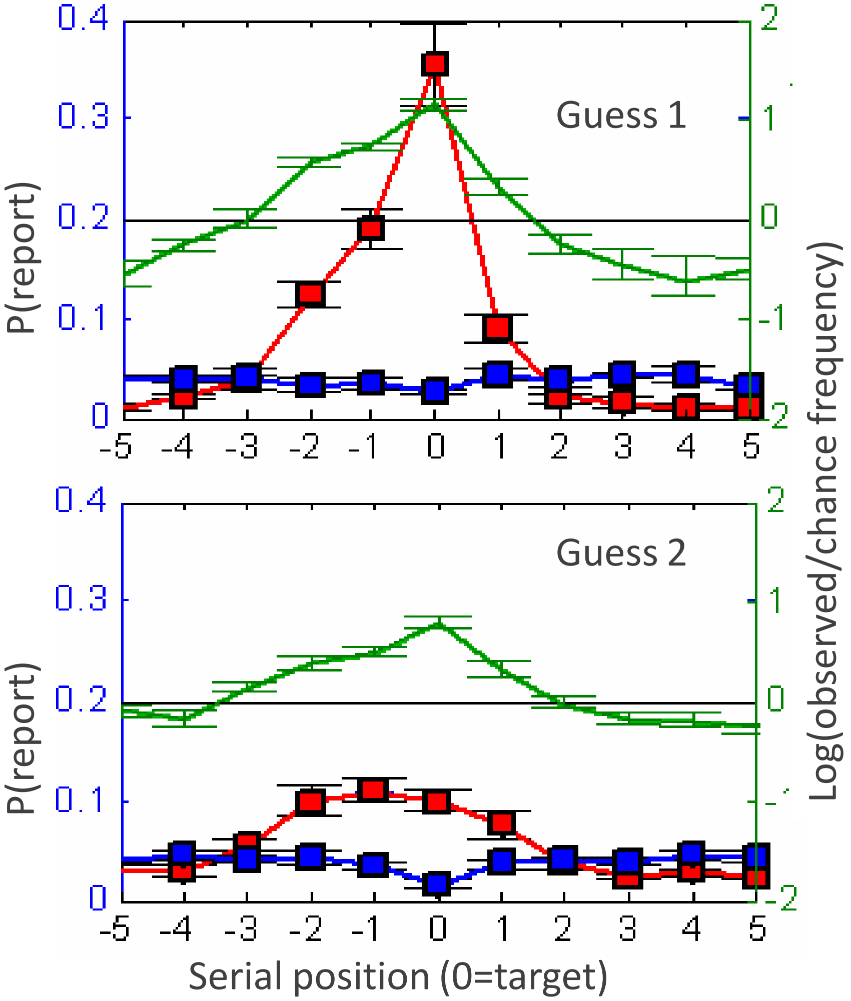


Figure by MIT OpenCourseWare.

# Multiple axes



# Two tradeoffs

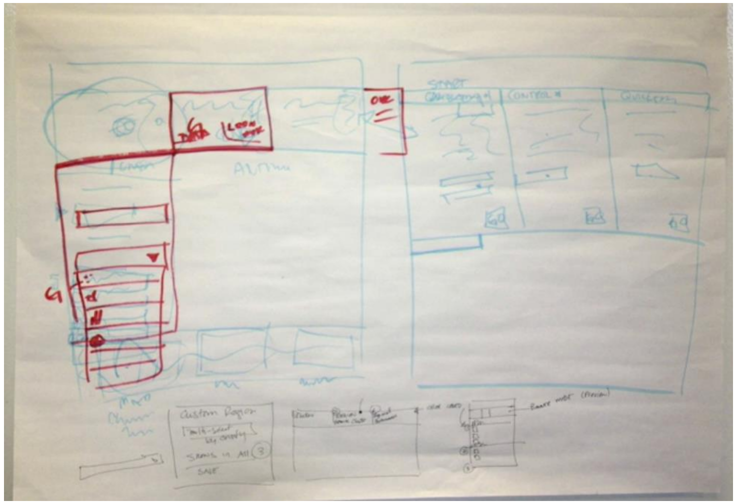
- Informativeness vs. readability
  - Too little information can conceal data
  - But too much information can be overwhelming
  - Possible solution: hierarchical organization?
- Data-centric vs. viewer-centric
  - Viewers are accustomed to certain types of visualization
  - But novel visualizations can be truer to data

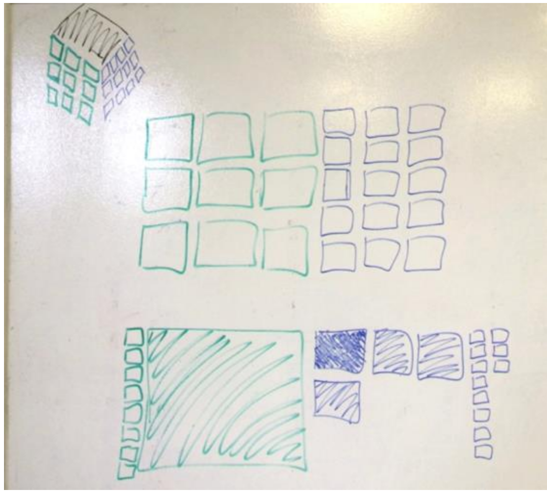
# To put it together …

So … there are many many options for visualising data (including a lot of fancy and interactive ones from the latest tools and libraries)

But let’s try to have some basic competency  on this:

- Accuracy is important, having a clear story to tell is important
- You need to be ready to do some basic data prep and pre analysis before visualisation
- Knowing the right paradigm (form) to use for the story
- Aware of your own limitation as ‘non-expert’ (visualisation is not easy)





Actually, a lot of experts recommend “sketching the idea out” with pen and paper.

# Data Visualization using Pandas/Matplotlib

- There are many excellent plotting libraries in Python and I recommend exploring more than one in order to create presentable graphics.
- In this course we are focusing on the Matplotlib library. It is the foundation for many other plotting libraries and plotting support in higher-level libraries such as Pandas.
- Don’t get confused with Matplotlib’s many ways of plotting the same thing. Pandas is our access point

# Data Visualization in Python

- **Matplotlib:** low level, provides lots of freedom
- **Pandas Visualization:** easy to use interface, built on Matplotlib
- **Seaborn:** high-level interface, great default styles
- **ggplot:** based on R’s ggplot2, uses Grammar of Graphics
- **Plotly:** can create interactive plots

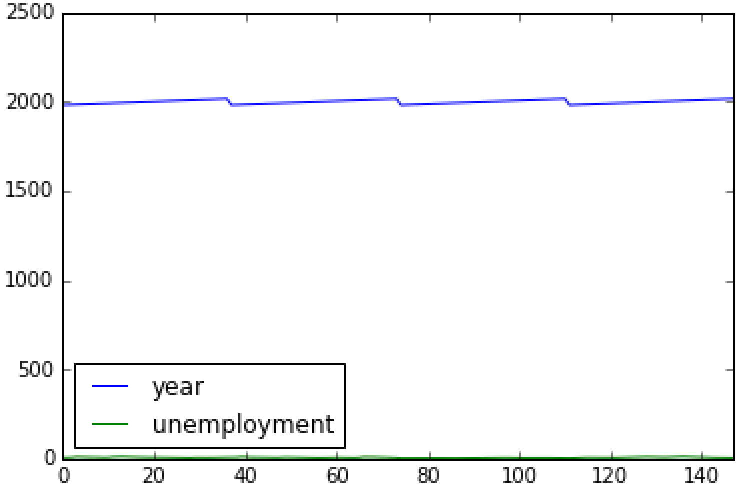
# Matplotlib and Dataframes

- Under the hood, pandas plots graphs with the matplotlib library. This is usually pretty convenient since it allows you to just .plot your graphs.
- When you use .plot on a dataframe, you sometimes pass things to it and sometimes you don't.
- .plot plots the index against every column
- .plot(x='col1') plots against a single specific column
- .plot(x='col1', y='col2') plots one specific column against another specific column

# Matplotlib and Dataframes

| | country | year | unemployment |
|---|---|---|---|
| 35 | Australia | 2015 | 6.063658 |
| 36 | Australia | 2016 | 5.723454 |
| 37 | USA | 1980 | 7.141667 |
| 38 | USA | 1981 | 7.600000 |
| 39 | USA | 1982 | 9.708333 |

If you use: df.plot()



http://jonathansoma.com/lede/algorithms-2017/classes/fuzziness-matplotlib/understand-df-plot-in-pandas/

# Matplotlib and Dataframes

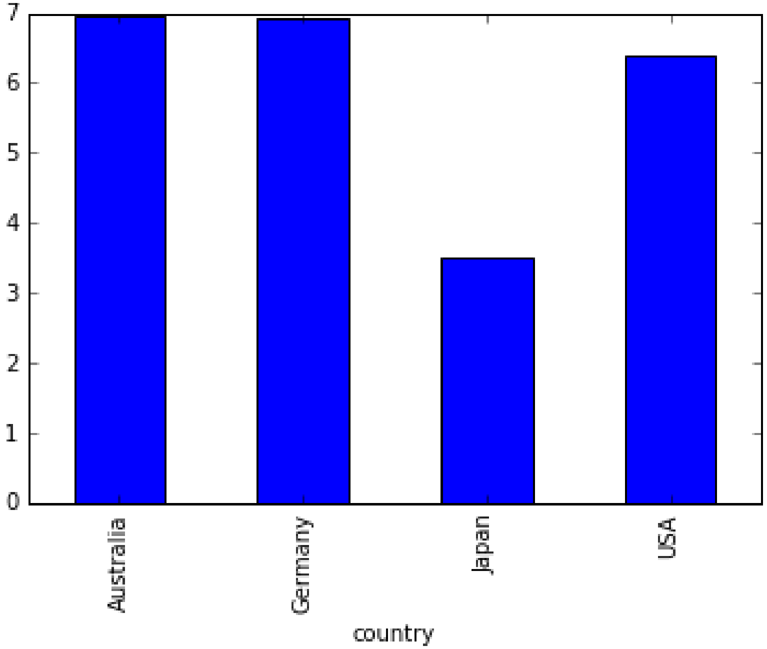
- The major use cases for .plot() is when you have a meaningful index, which usually happens in two situations:
    - You’ve just done a .value_counts() or a .groupby()
    - You’ve used .set_index, probably with dates

# Matplotlib and Dataframes

- So if you do:

df.groupby("country")['unemployment'].mean().plot(kind ='bar’)

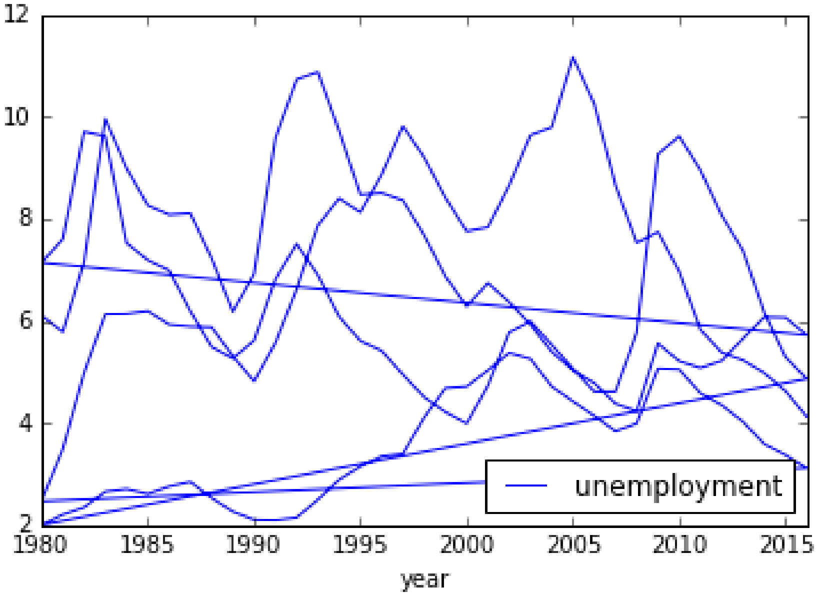
You’ll get:



# Matplotlib and Dataframes

- What about  (.plot(x='col1', y='col2’))
- Let’s try it for the same data before:

df.plot(x='year', y='unemployment')

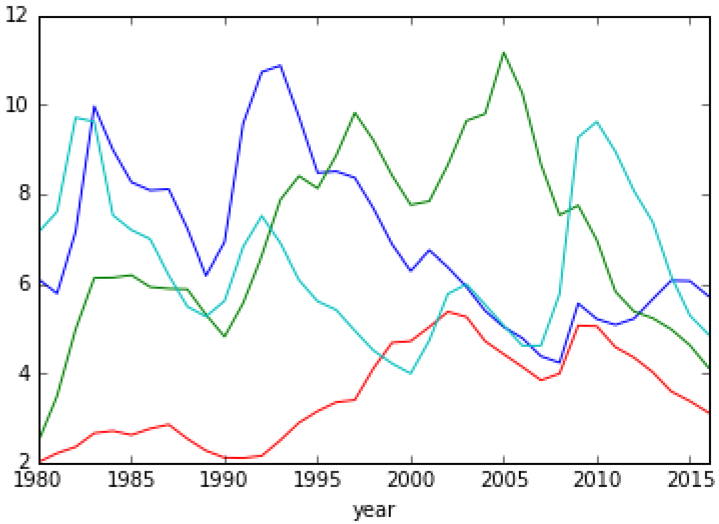


Talk about connected

# Matplotlib and Dataframes

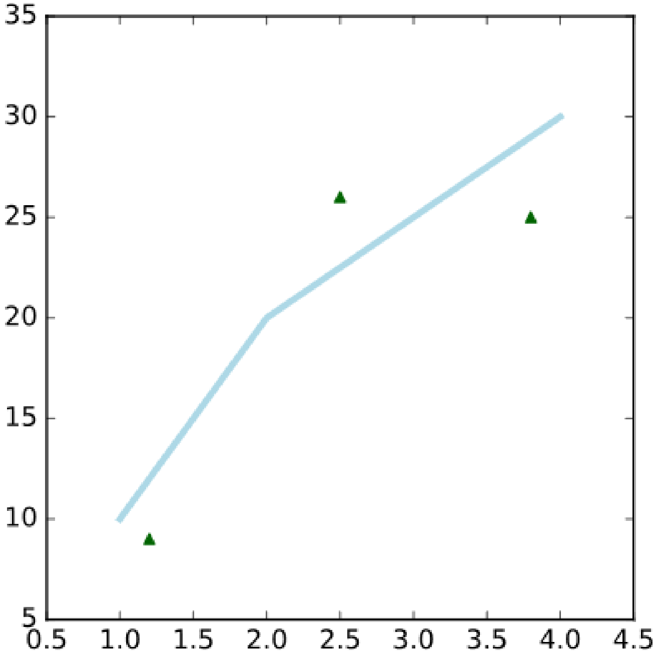
- Groupby to do it right.
- Create a single graph

```
fig, ax = plt.subplots()

df.groupby('country').plot(x='year', y='unemployment', ax=ax, legend=False)
```



# Matplotlib without dataframes

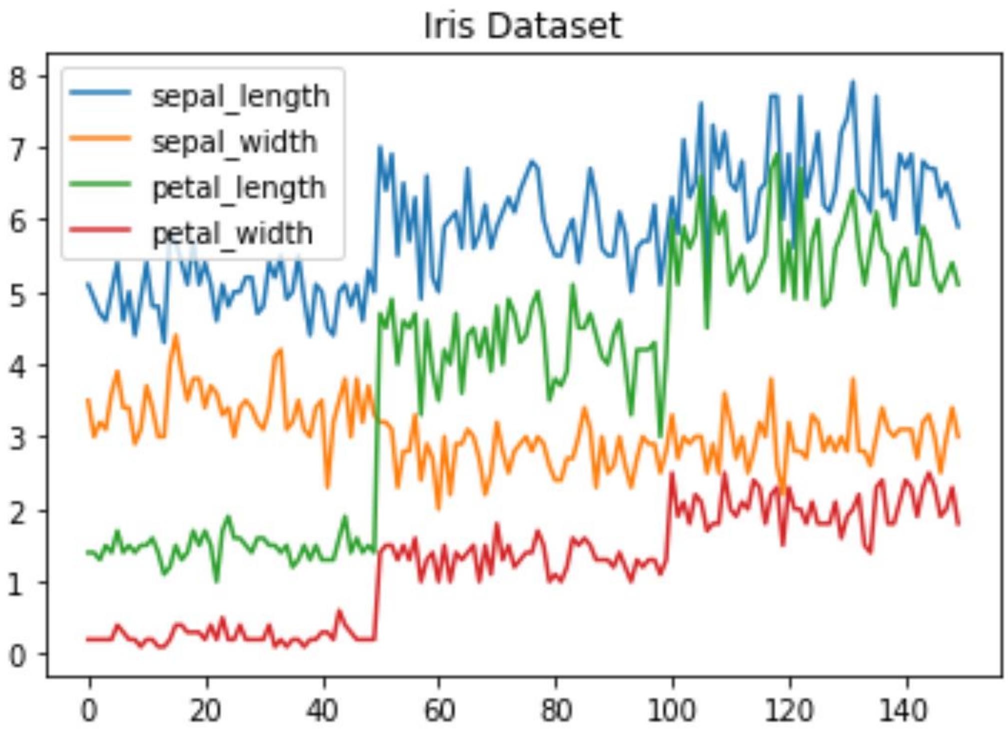
```
import matplotlib.pyplot as plt

plt.plot([1, 2, 3, 4], [10, 20, 25, 30], color='lightblue', linewidth=3)

plt.scatter([0.3, 3.8, 1.2, 2.5], [11, 25, 9, 26], color='darkgreen', marker='^’)

plt.xlim(0.5, 4.5)

plt.show()
```



# Matplotlib with/without Pandas

**With Pandas**

```
iris.drop(['class'], axis=1).plot.line(title='Iris Dataset')
```



**Without Pandas**

```
# get columns to plot
columns = iris.columns.drop(['class'])
# create x data
x_data = range(0, iris.shape[0])
# create figure and axis
fig, ax = plt.subplots()
# plot each column
for column in columns:
    ax.plot(x_data, iris[column])
# set title and legend
ax.set_title('Iris Dataset')
ax.legend()
```

# Matplotlib Conclusion

- Many details and scattered around documentation
- Stackoverflow is King
- DO YOUR Activities

# Useful Read

- Book: the Functional Art by Alberto Cairo (Chapter 1,2, and 3)
- http://jonathansoma.com/lede/algorithms-2017/classes/fuzziness-matplotlib/how-pandas-uses-matplotlib-plus-figures-axes-and-subplots/
- https://pythonspot.com/visualize-data-with-pandas/