DATA X

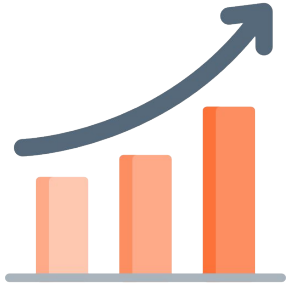
Data Visualization

Berkeley SCET

Created By:
Joshua Rafael Sanchez
joshuarafael@berkeley.edu

Module Structure

- Notebooks
- Slideshow
- Homework
- References



Part 1

Basic Visuals | Matplotlib, Seaborn

Basic Visualization Concepts, Introduction and Comparison b/t Matplotlib and Seaborn Python Libraries in Jupyter Notebook.

Part 2

Interactive Visuals | Plotly, Bokeh, Tableau, etc.

Deeper insights into more interactive and fun data visualization functions. Introduction to Plotly, Bokeh and Tableau.

Berkeley SCET

Icons made by Freepik from www.flaticon.com.

Table of Contents

(Note: Click on hyperlinks to go to different parts of the slides.)

0. About/Intro

1. Matplotlib

- About Matplotlib
- Installing Matplotlib
- Object Hierarchy
- Functional/MATLAB Approach (w/ ex)
- Object-Oriented Approach (w/ ex)

2. Seaborn

- About Seaborn
- Installing Seaborn
- Theme Adjustments (w/ ex)

3. Plotly

- About Plotly
- Installing Plotly
- Using Plotly Offline or Online
- Plotly Examples
- Plotly Alternatives:
  - Bokeh (w/ ex)
  - D3.js

4. Tableau

- About Tableau
- Tableau Desktop
- No-Code Visualization Tools
- Visualization Comparison

5. References

- Links to Notebooks
- References Cited

Data Visualization

Data-X: Applied Data Ventures

Sutardja Center at UC Berkeley

Berkeley SCET

What is data visualization?

Data visualization is the graphical representation of information and data.

What makes for effective data visualization?

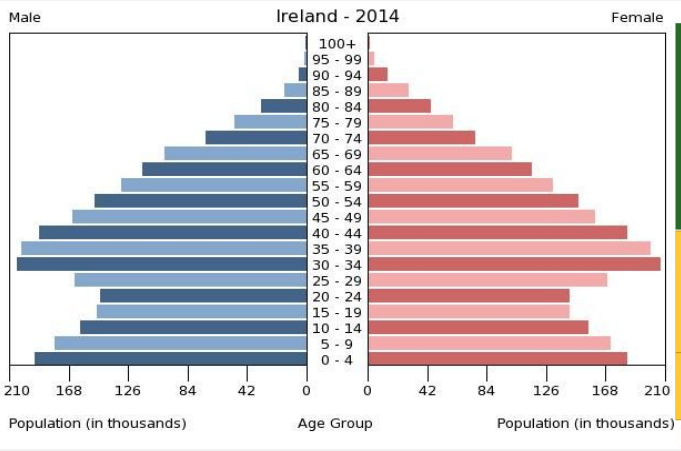
Visualization **transforms data into images effectively and accurately** represent information about the data.

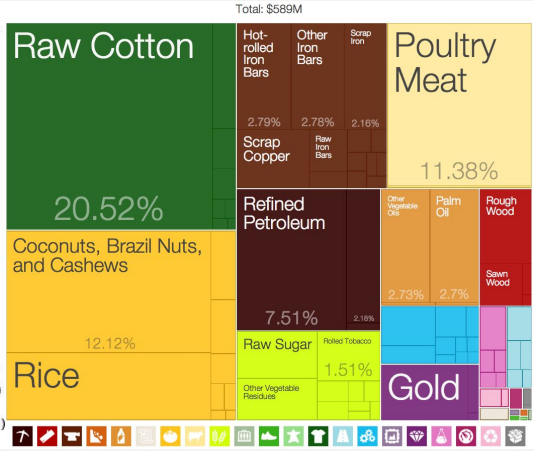
What are the advantages of data visualization?

Makes for easier **interpretation of patterns and trends** as opposed to looking at data in a tabular/spreadsheet format.

Examples of Data Visualizations

Left to Right: John Snow's 1854 Cholera Outbreak Map, Demographic Gender Breakdown, Government Budget Treemap of Benin





Berkeley SCET

About Data Visualization

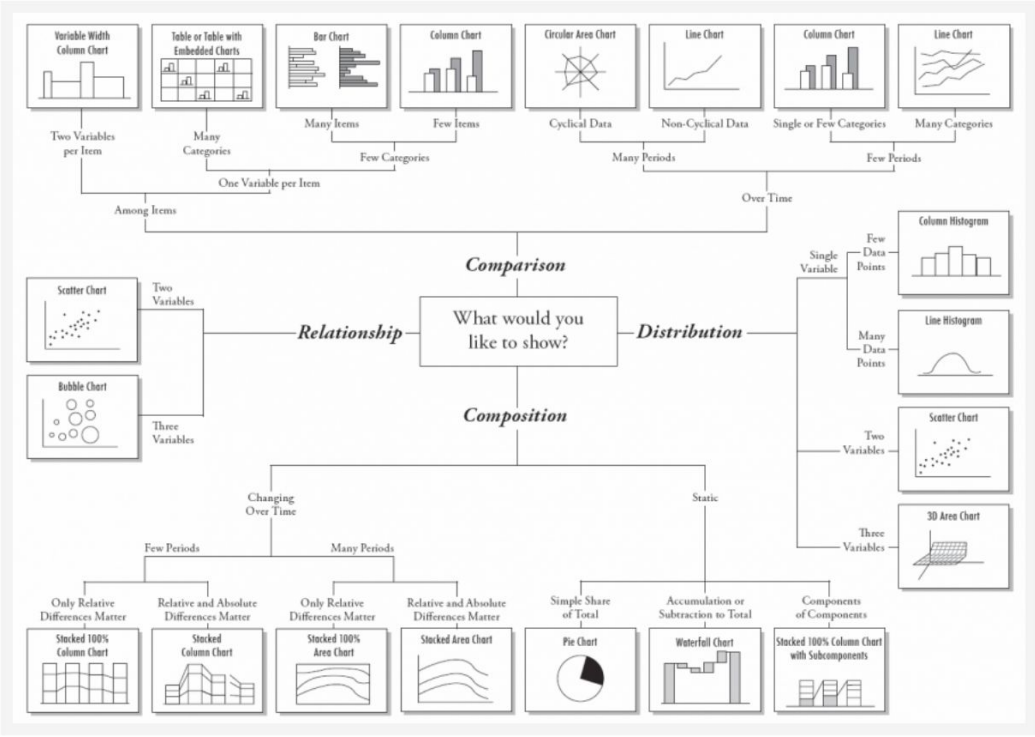
Painting a Picture of Data Visualization:

- Oxford English Dictionary Definition, 1989: To form a mental image, picture of (something not present or visible to the sight, or of an abstraction); to make visible to the mind or imagination
- There are 3 goals: To **explore** data, to **analyze** data, and/or to **present** data.

Question: What Would You Like to Show?

- Relationships between variables
- Composition of the data over time
- Distribution of variable(s) in data
- Comparison of data with relation to time, variables, categories, etc.

About Data Visualization



Berkeley SCET

Matplotlib

matplotlib.org/gallery

Matplotlib - About

About Matplotlib:

- Matplotlib is a comprehensive library for creating static, animated and interactive visualizations in Python.
- Usage: Matplotlib/Pandas is mostly used for quick plotting of Pandas DataFrames and time series analysis.

Pros and Cons of Matplotlib:

- Pro: Easy to setup and use.
- Pro: Very customizable.
- Con: Visual presentation tends to be simple compared to other tools.

Matplotlib - Installation

Installing Matplotlib should be straightforward. Sample code for installing packages:

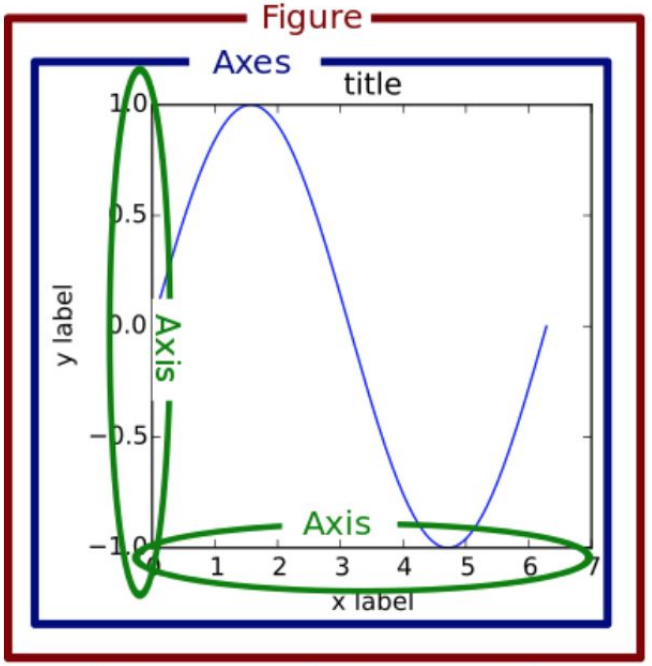
In [167]:

Slide Type Sub-Slide

```
# Standard way to import matplotlib from scratch: import matplotlib.pyplot as plt
# you might also see import pylab, however the procedure below is preferred

import numpy as np
import pandas as pd
import matplotlib
import IPython
import matplotlib as mpl
import matplotlib.pyplot as plt  # always import pyplot module as plt (standard)

%matplotlib inline
```

Matplotlib - Object Hierarchy

- **Figure**: Outermost container for a Matplotlib graphic. Can contain multiple Axes objects.
- **Axes**: Actual plots. Contain smaller objects (tick marks, individual lines, etc.)
- **Artist**: Everything that is seen on the figure is an artist.



https://realpython.com/python-matplotlib-guide/

Matplotlib - 2 Approaches to Plotting

1. ***Functional/MATLAB Approach (Non-Pythonic)***
   - Most common way of Matplotlib.
   - Pro: Easy approach for interactive use.
   - Con- Not pythonic: Relies on global functions (where variables are declared outside of functions) and displays global figures.

2. ***Object-Oriented Approach (Pythonic)***
   - **Recommended** way to use Matplotlib.
   - Pro: Pythonic is object-oriented (you can build plots explicitly using methods of the figure and the classes it contains.

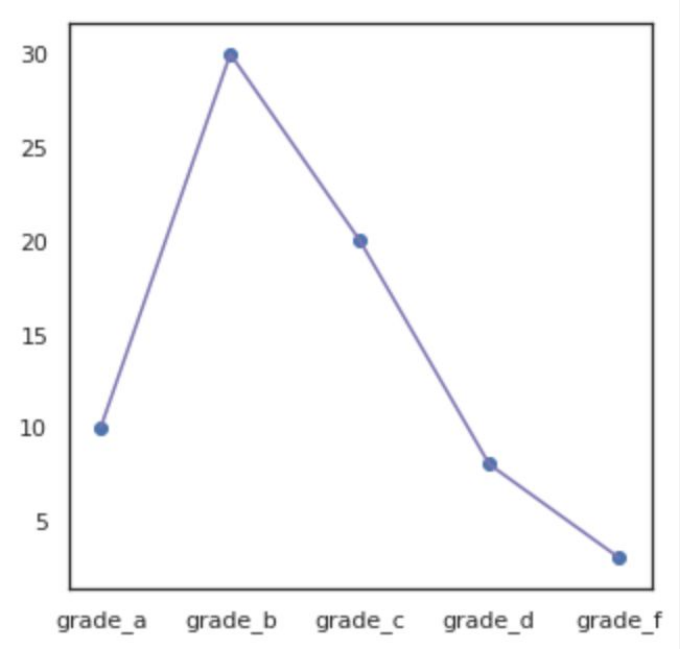
Matplotlib - Non-Pythonic Example

Example: Combining Line & Scatter Plots From Categorical Variables

In [179]:

Slide Type Sub-Slide

```
names = ['grade_a', 'grade_b', 'grade_c',
         'grade_d', 'grade_f']
values = [10, 30, 20, 8, 3]

plt.plot(names,values, 'm-')
plt.scatter(names, values);
```



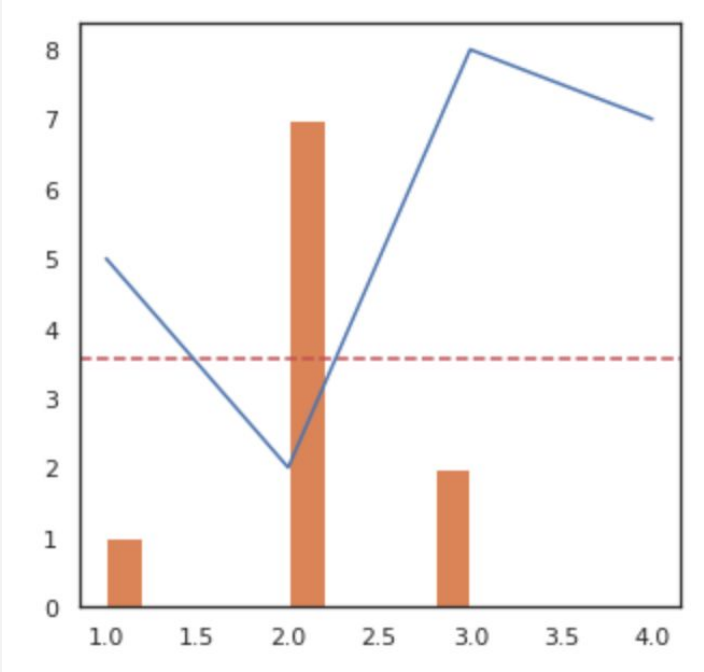
Matplotlib - Pythonic Example

Example: Simple Line Plot & Bar Plot

In [185]:

Slide Type Sub-Slide

```
# f is the canvas object, can contain several plots
# i.e. axes objects (p)
f, ax = plt.subplots() # returns tuple:

group_mean = np.mean(values)/4

# Add a horizontal line denoting average
ax.axhline(group_mean, ls='--', color='r')

# data as parameters
ax.plot([1,2,3,4],[5,2,8,7]);
ax.hist(np.random.randint(1,4,10));
```



Seaborn

seaborn.pydata.org

Seaborn - About

About Seaborn:

- Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.
- Usage: Those who want to create amplified data visuals, especially in color.

Seaborn's Pros and Cons:

- Pro: Includes higher level interfaces and settings than does Matplotlib
- Pro: Relatively simple to use, just like Matplotlib.
- Pro: Easier to use when working with Dataframes.
- Con: Like Matplotlib, data visualization seems to be simpler than other tools.

Seaborn - Installation

Installing Seaborn should also be straightforward. Sample code:

In [2]:

Slide Type Sub-Slide

```
from mpl_toolkits.mplot3d import Axes3D
from sklearn.decomposition import PCA
from sklearn import datasets
import seaborn as sns
import pandas as pd
```

Seaborn - Theme Adjustments

Theme Design- Setting Style:

- Use the five built-in themes to style the **figure/background of plots**:
  - Grids: `darkgrid`, `whitegrid`
  - Colors: `dark`, `white`, `ticks`.

Setting Scale:

- Use the four scaling plot presets to customize the **size of the plot**:
  - In order of relative size: `paper`, `notebook`, `talk`, `poster`.

Setting Fonts and Line Widths:

- How to change the **size of the text**:
  - Change the `font_scale` parameter for `sns.set_context()`.
- How to change the **line width of the text**:
  - Change the `rc` parameter for `sns.set_context()`.

Seaborn - Theme Adjustments w/ Examples

Let's look at the 5 built-in themes to style the figure (background of plots):

- Grids: `darkgrid`, `whitegrid`
- Colors: `dark`, `white`, and `ticks`.

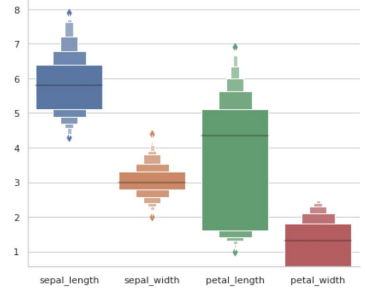
Consider examples using famous Iris Flower Data Set. Features of graphs:

- Left graph uses vertical bar plot w/ `whitegrid`, right graph uses swarm plot with `dark`.

In [148]:

Slide Type Sub-Slide

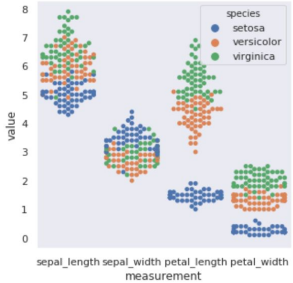
```
# set style locally
sns.set_style(style="whitegrid")

# plot categorical data
sns.catplot(data=df,        # dataframe
            kind='boxen',   # "point", "bar", "strip", "swarm", "box", "violin", or "boxen
            height=5,       # height of each facet
            aspect=1.2);    # width of each facet
```



In [149]:

Slide Type Sub-Slide

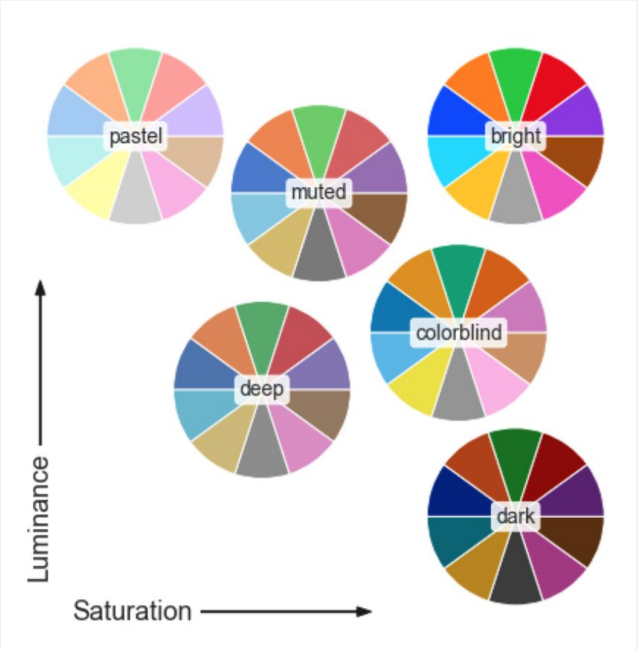
```
# Now with a swarmplot using  melted dataframe
sns.set_style(style="dark")

sns.swarmplot(x="measurement",
              y="value",        #
              hue="species",    # requires x,y
              data=iris);       # melted dataframe
```



Seaborn - Theme Adjustments: Color

Option 1- Default & Built-In Color Palettes:

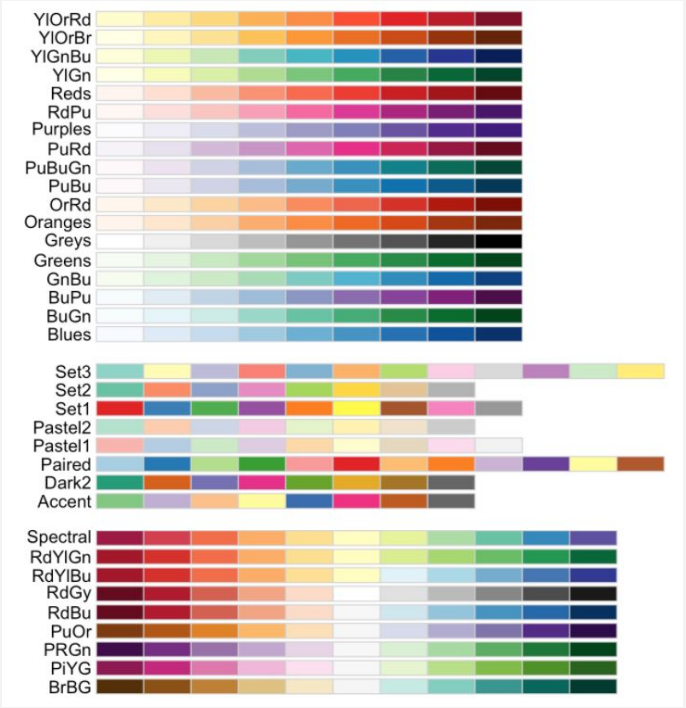
- About: Seaborn has six variations of its default color palette: deep, muted, pastel, bright, dark and colorblind.
- How to use: Use `sns.color_palette()` or `sns.set_palette()` for individual plots.
    - To set a color palette for all plots, use `sns.set(*args)`.



Seaborn - Theme Adjustments: Color

Option 2- Color Brewer Palettes:

- About: Created from the research of cartographer Cindy Brewer, these color palettes are specifically chosen as to be easy to interpret ordered categories.
- How to use: Use `sns.color_palette()` or `sns.set_palette()` for individual plots.
    - To set a color palette for all plots, use `sns.set(*args)`.



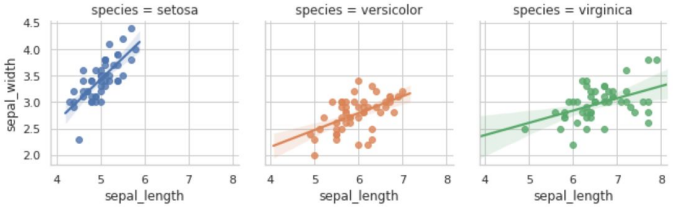
Seaborn - Theme Adjustments: Color Examples

Left image: Code and resulting plot using default & built-in color palettes.
Right image: Code and resulting plot using a Color Brewer palette.

In [153]:

Slide Type: Sub-Slide

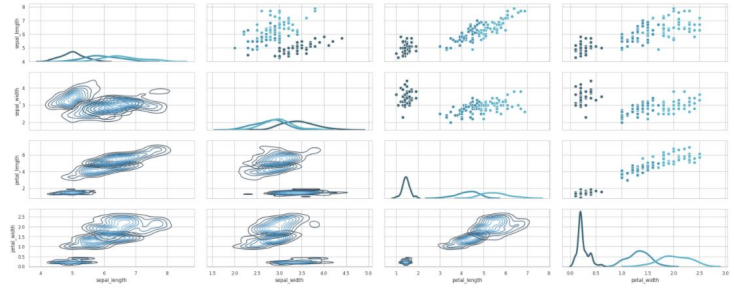
```
# set palette
sns.color_palette('dark')

# set style
sns.set_style('whitegrid')

# initialize a facetgrid object
g = sns.FacetGrid(df, col='species', hue='species', col_wrap=4)
g.map(sns.regplot, 'sepal_length', 'sepal_width');
```



In [159]:

Slide Type: Sub-Slide

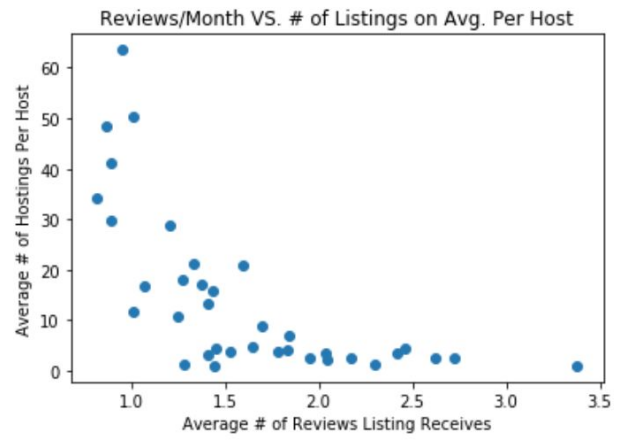
```
# initialize pairgrid object and set design and color
g = sns.PairGrid(data=df,
                 hue='species',
                 palette = sns.color_palette("GnBu_d", 4),
                 height=2.0,
                 aspect=2.5,
                 despine=False)

# set order of plots
g.map_upper(plt.scatter, edgecolor="w", s=40)
g.map_lower(sns.kdeplot, cmap="Blues_d")
g.map_diag(sns.kdeplot, lw=3, legend=False);
```

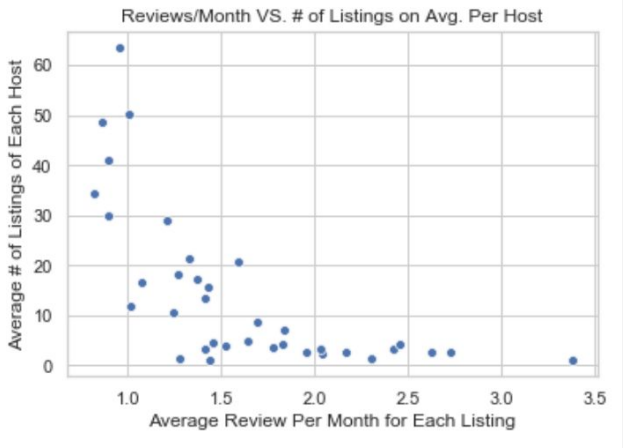


Berkeley SCET

Matplotlib vs. Seaborn

- Visuals
- Options
- Interactivity





Plotly
plotly.com/python

Plotly - About

About Plotly:

- From website: Plotly is an interactive, open-source plotting library that supports over 40 unique chart types.
- Usage: Plotly is advantageous for those who want an interactive environment which many use cases, ranging from statistics to finance to geography and more.

Pros and Cons of Plotly:

- Pro: Make **beautiful, interactive, exportable** figures in just a few lines of code.
- Pro: Much more interactive & visually flexible than Matplotlib or Seaborn.
- Con: Confusing initial setup to use Plotly without an online account, and lots of code to write.
- Con: Out-of-date documentation and the large range of Plotly tools (Chart Studio, Express, etc.) make it hard to keep up.

Plotly - Installing

Installing Plotly Offline: (if you want to host locally on your own computer)

- Steps: You need to import packages and use commands:
  - Resource: Keep checking current version: Initialization for Online Plotting
  - Command to create standalone HTML: `plotly.offline.plot()`
  - Command to create plot in Jupyter Notebook: `plotly.offline.iplot()`

Installing Plotly Online: (use if you want to host graphs in plotly account)

- How to: You must create an account to run:
  1. Set up an account at plot.ly
  2. Get a User ID and API keys
  3. Sign keys into the account.

Plotly - Alternatives (Bokeh, D3.js)

Bokeh:

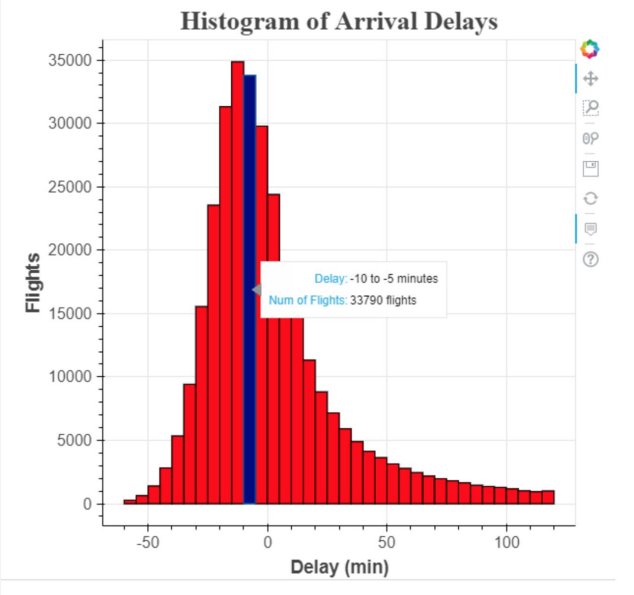
- Bokeh is an interactive visualization Python library.
- Provides elegant and concise construction of versatile graphics.
- Usage: Can be used in Jupyter Notebooks and can provide high-performance interactive charts and plots.

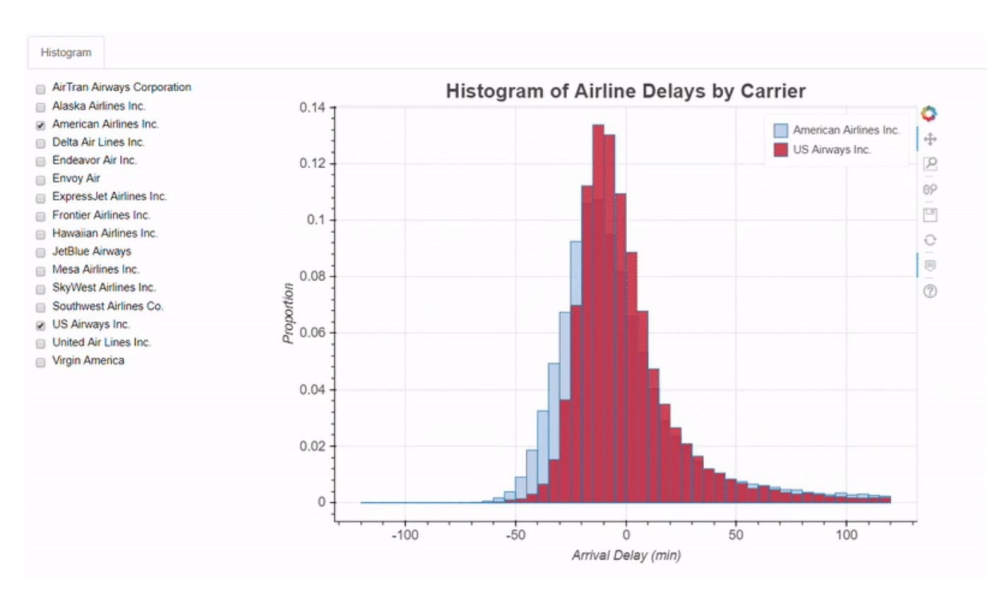
D3.js:

- D3.js (used with Flask) is a framework used with HTML, CSS, and Javascript together to create visualizations.
- Usage: Use D3.js build-in data-driven transitions for extra customization and elevated visualization for your data.
- Pro: Helps build type of framework you want (Plotly uses D3.js library, here you can use the D3.js library itself; open-source)
- Con: High learning curve; you need to learn HTML, CSS, Javascript

Bokeh - Example

Example of using Bokeh from article. Screenshots of interactive features that Bokeh offers:





Tableau

https://www.tableau.com/

Tableau: Intro & Setup

What Are Dashboards:

- Dashboards act as a data visualization tool where users can easily analyze trends and statistics. It can be a powerful way of communicating results of a Data Science project.
- Examples: Dash by Plotly, Bokeh Dashboards, Google Data Studio, Tableau

About Tableau (Tableau Desktop):

- Pros: Makes the charts and interface almost seamlessly.
- Con: Getting used to the interface and functions.
- Con: Data cleaning/pre-processing easier in Python.

Setting up:

- 1-year free trial of Tableau Desktop for Students. (Paid differs by individual vs organization.)
- Tableau Public (create separate account); share data visualizations with global community.
- Introductory videos are a great resource; robust and go through examples in detail.

Tableau - Tableau Desktop (for Students)

Go to this link to try out a trial: https://www.tableau.com/academic/students

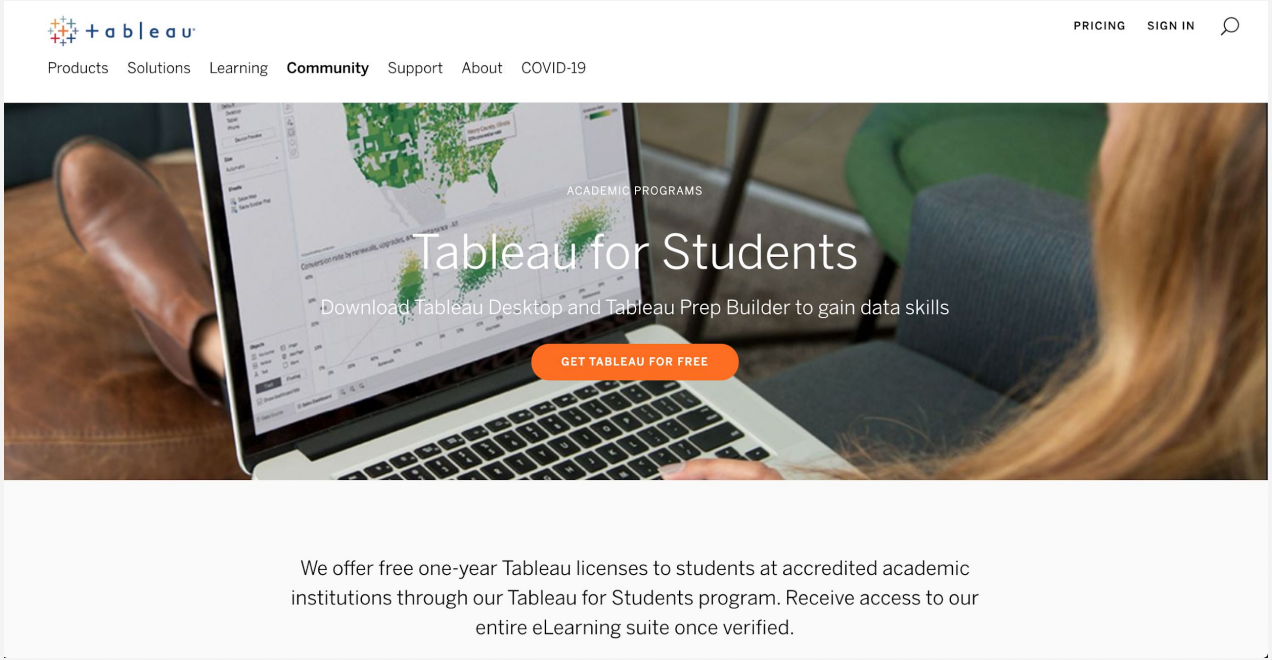
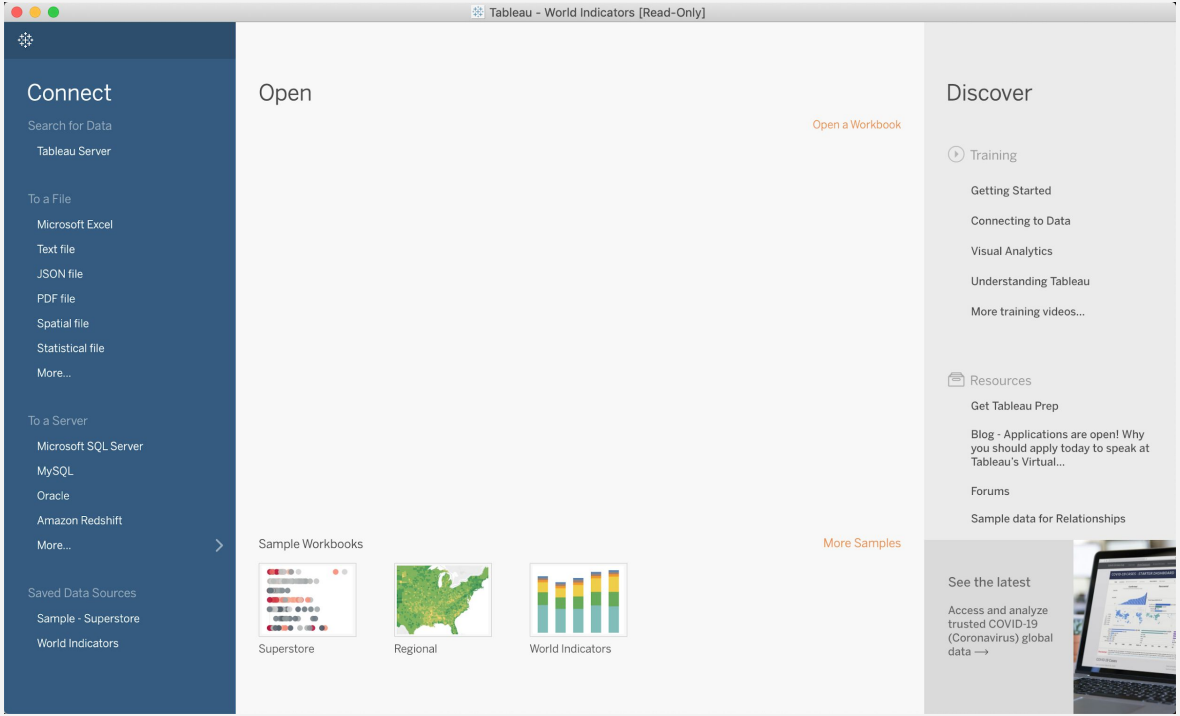


Tableau - Tableau Desktop (for Students)

When you download the Tableau Desktop Application (MacBook Pro):



Explore: No-Code Visualization Tools

Infogram: | https://infogram.com/app/

- Web-based visualization environment; **infographic environment.**
- Multiple PDF/PNG or HTML-based templates; interactivity built-in.
- Paid version offers: Engagement analytics, team collaboration, consistent product branding.

Flourish: | https://flourish.studio/examples/

- Another web-based visualization environment.
- Interest: Interface is pretty straightforward, and visualizations can be really interactive.
- Note: Best for **spreadsheet junkies!**

Datawrapper: | https://www.datawrapper.de/

- Web-based visualization and map creation environment.
- **Niche service,** offers some powerful capabilities.
- Fact: Interesting workflow.

Visualization Tools Comparison

	Data import & usage	Viz options & customization	Free/paid features	More or less technical?
tableau	- Can import from many data types. - Robust manipulation.	- **Many graph options.** - Experienced users understand benefit.	- Tableau Public - Tableau Desktop (1-Year free trial student)	- More technical due to interface and multitude of options.
infogr.am	- Can import from some data types. - Some manipulation.	- Many **infographic** visual options. - Drag & drop interface.	- Free w/ account; - Make publicly available PDF, PNG or HTML	- Less technical - No code; interface accessible to all.
Flourish	- Import from Microsoft Excel, CSV, JSON. - Some manipulation.	- **Graph, infographic and slide** options. - Straightforward editing interface.	- Free w/ account; - Embed, PDF, PNG, or HTML.	- Less technical - No code; interface accessible to all.
Datawrapper	- Import from multiple sources. - Minimal manipulation.	- Static graph options. - **Streamlined process** of creating visualizations	- Free (no account need) - PDF, PNG, or HTML	- Less technical. - Frequently used by journalists/newspapers.

References

Data Visualization - References

Color Palette & Logos

Berkeley Blue
#003262

California Gold
#FDB515

Black
#000000

Dark Gray
#434343

Light Gray
#efefefff

DATA X

Berkeley SCET

Berkeley SCET

Font: **Helvetica Neue**
Size of Titles: **28**

Diagram Styleguide

Ticker Template (Copy and Paste)

Ticker Ticker Ticker Ticker Ticker

Example Title

“Replace squares with icons”

Header 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vitae tincidunt dolor.

Header 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vitae tincidunt dolor.

Header 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vitae tincidunt dolor.

Header 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vitae tincidunt dolor.

Header 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vitae tincidunt dolor.

Header 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vitae tincidunt dolor.