Data Visualization in Data Science

Dr. Ha Nguyen
Ambyint - Canada

Week 7 - Agenda

1. Introduction to Visualization Tools
2. Basic Visualization Tools
3. Specialized Visualization Tools
4. Advanced Visualization Tools
5. Creating Maps and Visualizing Geospatial Data

Week 7 - Agenda

1. Introduction to Visualization Tools
2. Basic Visualization Tools
3. Specialized Visualization Tools
4. Advanced Visualization Tools
5. Creating Maps and Visualizing Geospatial Data

Why Data Visualization

1. For exploratory data analysis
2. Communicate data clearly
3. Share unbiased representation of data
4. Use them to support recommendations to different stakeholders

Always remember:

LESS is more attractive, effective, and impactive

Why Data Visualization

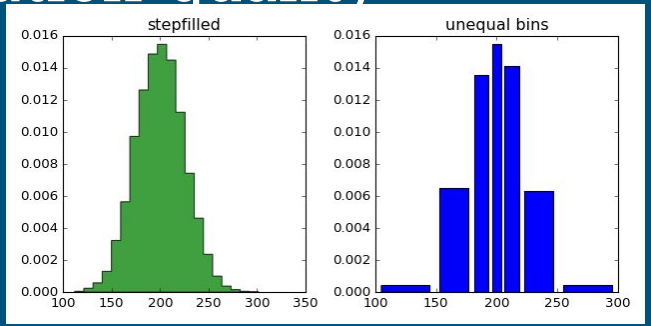
“One of the best but also more challenging ways to get your insights across is to visualize them: that way, you can more easily identify patterns, grasp difficult concepts or draw the attention to key elements”

Introduction to Visualization Tools

1. Visualization Tools
    a. Matplotlib
    b. Seaborn
    c. Folium
2. Best practices
    a. Less is more effective
    b. Less is more attractive
    c. Less is more impactive

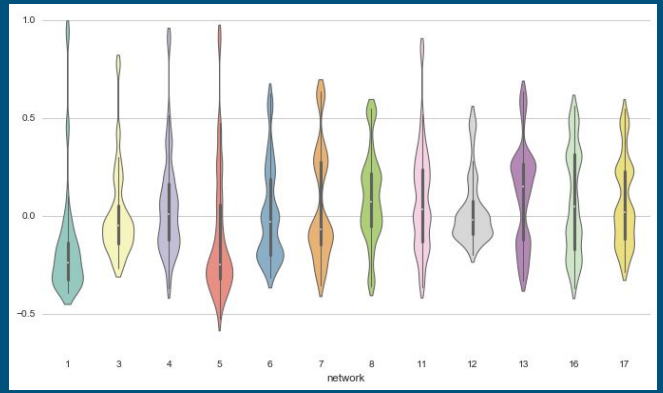
matplotlib

1. Initial release: 2003
2. Python data visualization tool
3. Designed to closely ensemble MATLAB
4. Many other tools built on top of it: pandas, Seaborn
5. Not very useful for creating publication-quality charts quickly and easily



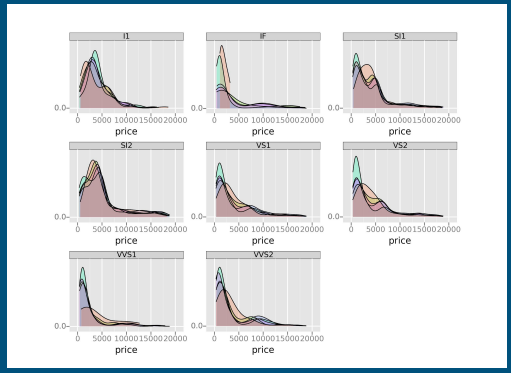
seaborn

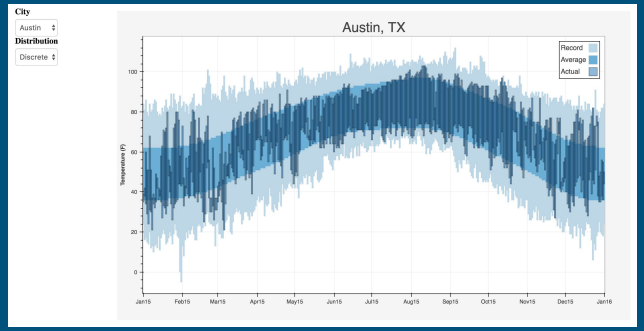
1. Built on top of matplotlib: “Or, as Michael Waskom says in the “Introduction to Seaborn”: “If matplotlib “tries to make easy things easy and hard things possible”, seaborn tries to make a well-defined set of hard things easy too.”
2. Harnesses the power of matplotlib to create beautiful charts in a few lines of code

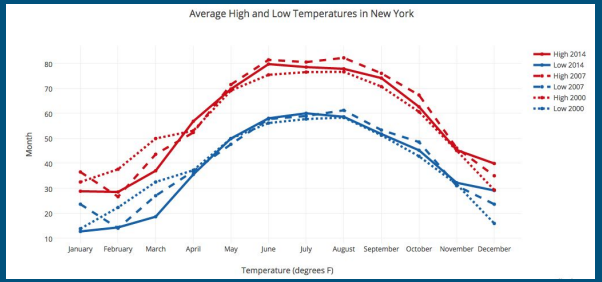


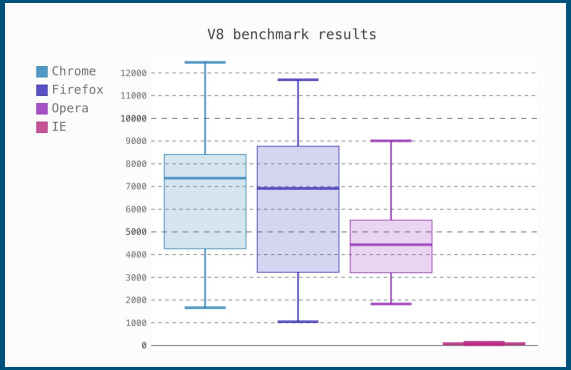
Other visualization tools

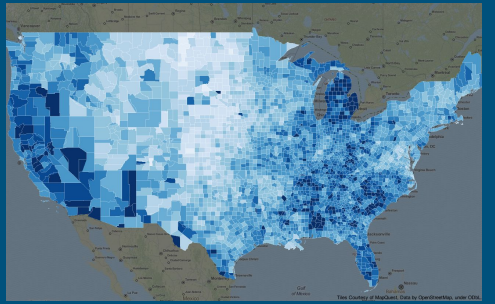
1. ggplot
2. Plotly
3. Geoplotlib
4. Bokeh
5. pygal
6. Gleam
7. missingno
8. Leather











Introduction to Visualization Tools - Datasets

http://www.un.org/en/development/desa/population/migration/data/empirical2/migrationflows.shtml

- The population division of the United Nations compiled immigration data pertaining to 45 countries. The data consist of the total number of immigrants from all over the world to each of the 45 countries as well as other metadata pertaining to the immigrants countries of origin.
- Focus on immigration to Canada and we will work primarily with the data set involving immigration to the great white north

Introduction to Visualization Tools - Datasets

1. View dataset in Excel
2. Read data into

   Pandas Dataframe
3. View data

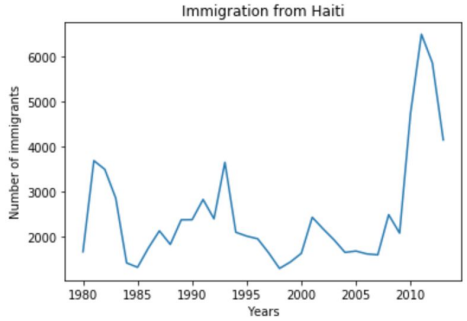
```
In [1]: import numpy as np
        import pandas as pd

In [2]: #!pip install xlrd # to read data into pd from Excel spreadsheet

In [7]: df_can = pd.read_excel('Canada.xlsx',
                               sheet_name='Canada by Citizenship',
                               skiprows =range(20),
                               skipfooter=2)

In [8]: df_can.head(5)

Out[8]:
```

lame	AREA	AreaName	REG	RegName	DEV	DevName	1980	...	2004	2005	2006	2007	2008	2009	2010	2011	2012	20
istan	935	Asia	5501	Southern Asia	902	Developing regions	16	...	2978	3436	3009	2652	2111	1746	1758	2203	2635	20
ania	908	Europe	925	Southern Europe	901	Developed regions	1	...	1450	1223	856	702	560	716	561	539	620	6
geria	903	Africa	912	Northern Africa	902	Developing regions	80	...	3616	3626	4807	3623	4005	5393	4752	4325	3774	43
rican amoa	909	Oceania	957	Polynesia	902	Developing regions	0	...	0	0	1	0	0	0	0	0	0	
dorra	908	Europe	925	Southern Europe	901	Developed regions	0	...	0	0	1	1	0	0	0	0	1	

Introduction to Visualization Tools - Line

- The trend of Haitian immigrants to Canada from 1980 to 2013: Line plot
- Plot in the form of a series of data points connected by straight line segments

```
In [40]: haiti.index = haiti.index.map(int)
         haiti.plot(kind='line')

         plt.title('Immigration from Haiti')
         plt.ylabel('Number of immigrants')
         plt.xlabel('Years')

         plt.show()
```



Week 7 - Agenda

1. Introduction to Visualization Tools
2. Basic Visualization Tools
3. Specialized Visualization Tools
4. Advanced Visualization Tools
5. Creating Maps and Visualizing Geospatial Data

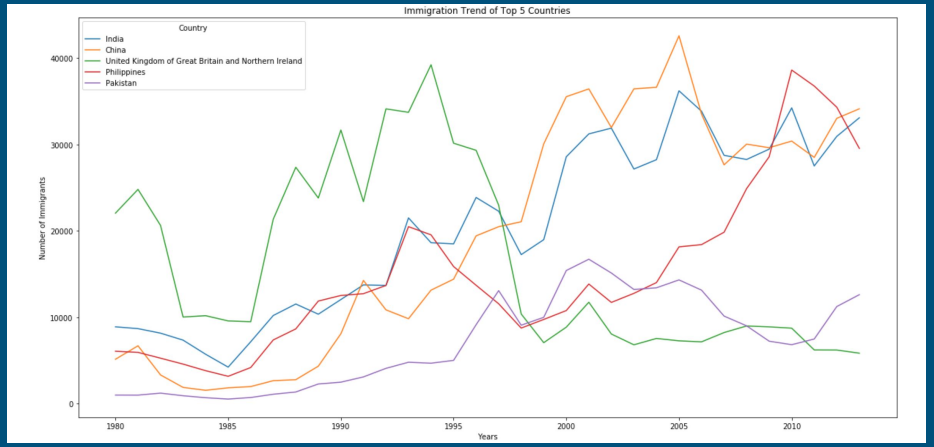
Basic Visualization Tools

1. Visualization Tools
    a. Area Plots
    b. Histograms
    c. Bar Charts

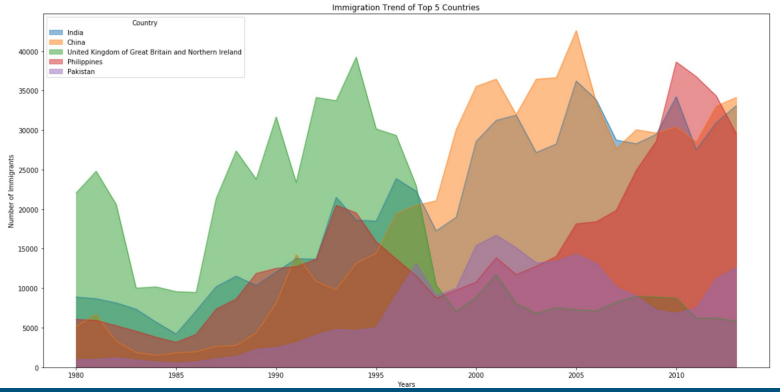
Basic Visualization Tools - Area Plots

- Depicts accumulated totals using numbers or percentages over time.
- Based on the line plot and is commonly used when trying to compare two or more quantities
- Visualize the top 5 countries that contributed the most immigrants to Canada from 1980 to 2013

Basic Visualization Tools - Area Plots



In [35]:
```
df_top5.index = df_top5.index.map(int) # let's change the index values of df_top5 to type integer for plotting
df_top5.plot(kind='area',
             stacked=False,
             figsize=(20, 10), # pass a tuple (x, y) size
             )

plt.title('Immigration Trend of Top 5 Countries')
plt.ylabel('Number of Immigrants')
plt.xlabel('Years')

plt.show()
```



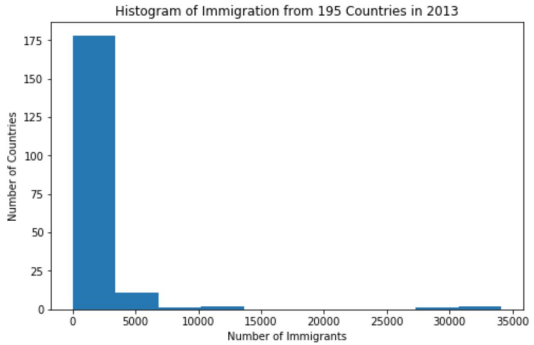
Basic Visualization Tools - Histogram Plots

- A histogram is a way of representing the *frequency* distribution of numeric dataset.
- Partitions the x-axis into *bins*, assigns each data point in our dataset to a bin, and then counts the number of data points that have been assigned to each bin.
- y-axis is the frequency or the number of data points in each bin.

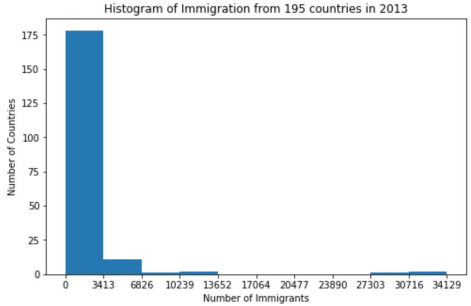
Basic Visualization Tools - Histogram Plots

Question: What is the frequency distribution of the number (population) of new immigrants from the various countries to Canada in 2013?

In [40]:
```
df_can['2013'].plot(kind='hist', figsize=(8, 5))

plt.title('Histogram of Immigration from 195 Countries in 2013') # add a title to the histogram
plt.ylabel('Number of Countries') # add y-label
plt.xlabel('Number of Immigrants') # add x-label

plt.show()
```




```
# 'bin_edges' is a list of bin intervals
count, bin_edges = np.histogram(df_can['2013'])

df_can['2013'].plot(kind='hist', figsize=(8, 5), xticks=bin_edges)

plt.title('Histogram of Immigration from 195 countries in 2013') # add a title to the histogram
plt.ylabel('Number of Countries') # add y-label
plt.xlabel('Number of Immigrants') # add x-label

plt.show()
```



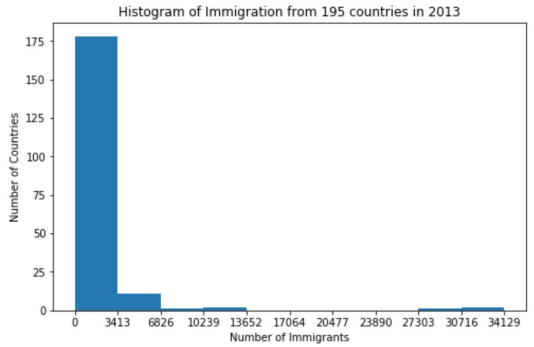
Basic Visualization Tools - Bar Charts

- Unlike a histogram, a bar chart also known as a bar graph is a type of plot where the length of each bar is proportional to the value of the item that it represents.
- It is commonly used to compare the values of a variable at a given point in time.

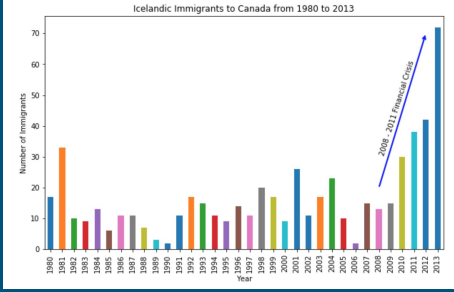
Basic Visualization Tools - Bar Charts

We're interested in visualizing in a discrete fashion how immigration from Iceland to Canada looked like from 1980 to 2013

```
# 'bin_edges' is a list of bin intervals
count, bin_edges = np.histogram(df_can['2013'])

df_can['2013'].plot(kind='hist', figsize=(8, 5), xticks=bin_edges)

plt.title('Histogram of Immigration from 195 countries in 2013') # add a title to the histogram
plt.ylabel('Number of Countries') # add y-label
plt.xlabel('Number of Immigrants') # add x-label

plt.show()
```




```
df_iceland.plot(kind='bar', figsize=(10, 6), rot=90)

plt.xlabel('Year')
plt.ylabel('Number of Immigrants')
plt.title('Icelandic Immigrants to Canada from 1980 to 2013')

# Annotate arrow
plt.annotate('',                      # s: str. will leave it blank for no text
             xy=(32, 70),             # place head of the arrow at point (year 2012 , pop 70)
             xytext=(28, 20),         # place base of the arrow at point (year 2008 , pop 20)
             xycoords='data',         # will use the coordinate system of the object being annotated
             arrowprops=dict(arrowstyle='->', connectionstyle='arc3', color='blue', lw=2)
            )

# Annotate Text
plt.annotate('2008 - 2011 Financial Crisis', # text to display
             xy=(28, 30),                    # start the text at at point (year 2008 , pop 30)
             rotation=72.5,                  # based on trial and error to match the arrow
             va='bottom',                    # want the text to be vertically 'bottom' aligned
             ha='left',                      # want the text to be horizontally 'left' algned.
            )

plt.show()
```



Week 7 - Agenda

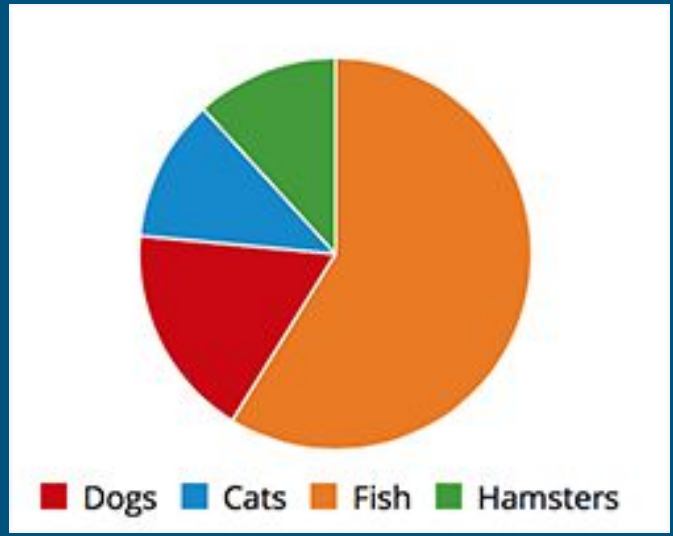
1. Introduction to Visualization Tools
2. Basic Visualization Tools
3. Specialized Visualization Tools
4. Advanced Visualization Tools
5. Creating Maps and Visualizing Geospatial Data

Specialized Visualization Tools

1. Visualization Tools
    a. Pie Charts
    b. Box Plots
    c. Scatter Plots

Specialized Visualization Tools - Pie Charts

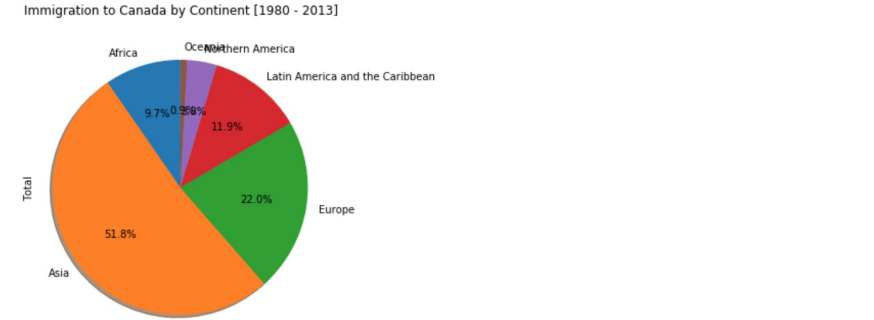
A pie chart is a circular statistical graphic divided into slices to illustrate numerical proportion



Specialized Visualization Tools - Pie Charts

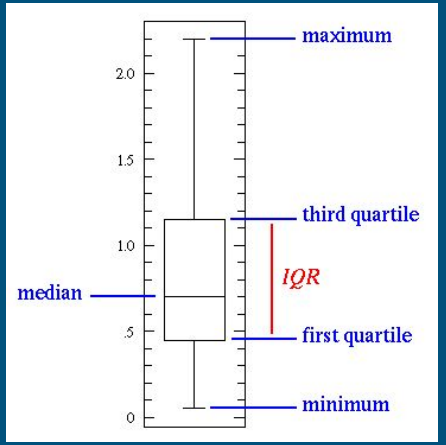
Let's use a pie chart to explore the proportion (percentage) of new immigrants grouped by continents for the entire time period from 1980 to 2013

```
# autopct create %, start angle represent starting point
df_continents['Total'].plot(kind='pie',
                            figsize=(5, 6),
                            autopct='%1.1f%%', # add in percentages
                            startangle=90,     # start angle 90° (Africa)
                            shadow=True,       # add shadow
                            )

plt.title('Immigration to Canada by Continent [1980 - 2013]')
plt.axis('equal') # Sets the pie chart to look like a circle.

plt.show()
```



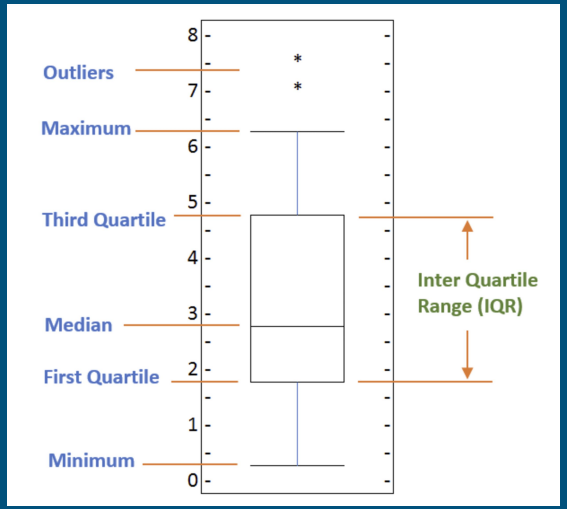
Specialized Visualization Tools - Box Plots

- A box plot is a way of statistically representing the distribution of given data through five main dimensions.
- The **first dimension** is minimum, which is the smallest number in the sorted data.
- The **second dimension** is first quartile, which is the point 25% of the way through the sorted data.



Specialized Visualization Tools - Box Plots

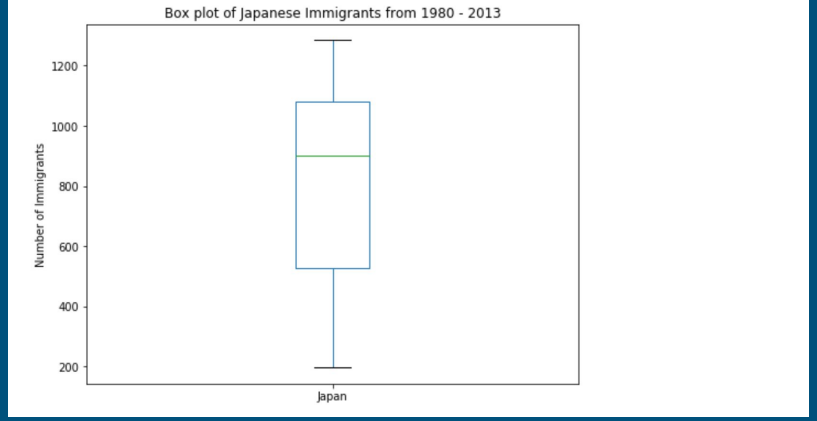
- The **fourth dimension** is third quartile, which is the point 75% of the way through the sorted data.
- The **final dimension** is maximum, which is the highest number in the sorted data.



Specialized Visualization Tools - Box Plots

Let's plot the box plot for the Japanese immigrants between 1980 - 2013.

```
df_japan.plot(kind='box', figsize=(8, 6))

plt.title('Box plot of Japanese Immigrants from 1980 - 2013')
plt.ylabel('Number of Immigrants')

plt.show()
```



Specialized Visualization Tools - Scatter Plots

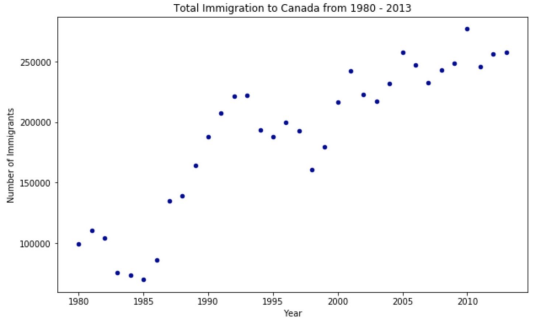
- A scatter plot is a type of plot that displays values pertaining to typically two variables against each other.
- Usually it is a dependent variable to be plotted against an independent variable in order to determine if any correlation between the two variables exists.

Specialized Visualization Tools - Scatter Plots

Let's visualize the trend of total immigration to Canada (all countries combined) for the years 1980 - 2013.

	year	total
0	1980	99137
1	1981	110563
2	1982	104271
3	1983	75550
4	1984	73417

```
df_tot.plot(kind='scatter', x='year', y='total', figsize=(10, 6), color='darkblue')

plt.title('Total Immigration to Canada from 1980 - 2013')
plt.xlabel('Year')
plt.ylabel('Number of Immigrants')

plt.show()
```



Week 7 - Agenda

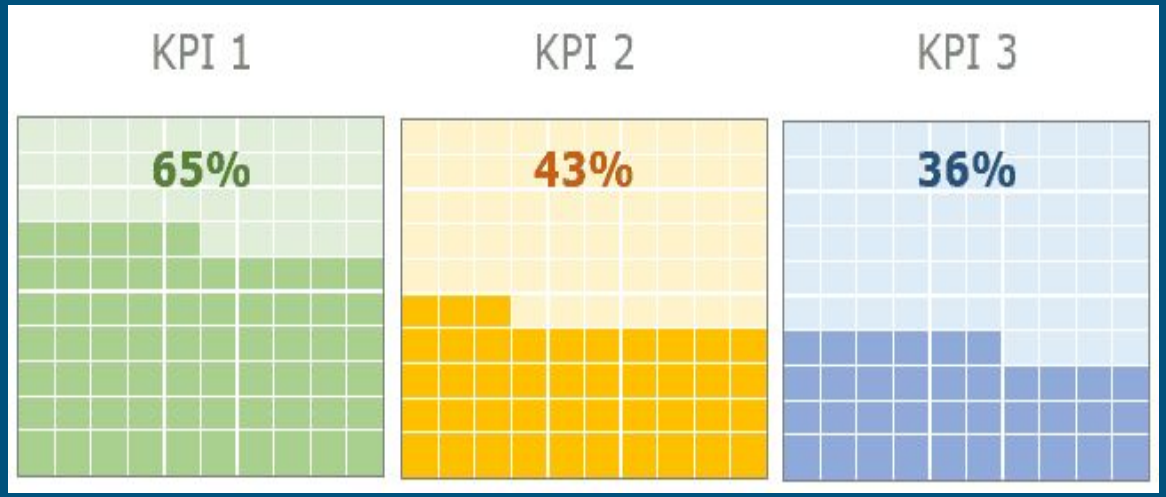
1. Introduction to Visualization Tools
2. Basic Visualization Tools
3. Specialized Visualization Tools
4. Advanced Visualization Tools
5. Creating Maps and Visualizing Geospatial Data

Advanced Visualization Tools

1. Visualization Tools
    a. Waffle Charts
    b. Word Clouds
    c. Seaborn and Regression Plots

Advanced Visualization Tools - Waffle Charts

A waffle chart is a great way to visualize data in relation to a whole or to highlight progress against a given threshold.



Advanced Visualization Tools - Word Clouds

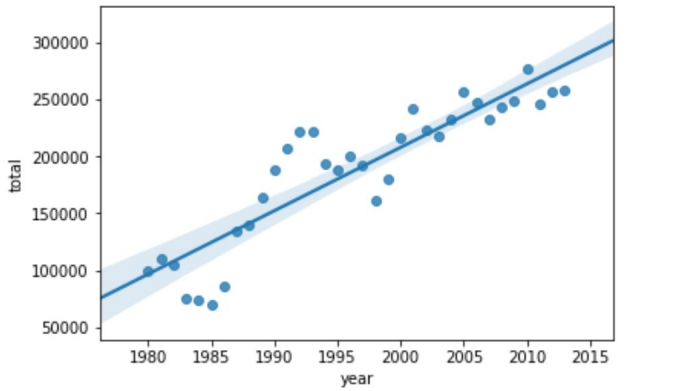
- A word cloud is simply a depiction of the importance of different words in the body of text.
- The more a specific word appears in a source of textual data the bigger and bolder it appears in the world cloud.

Advanced Visualization Tools - Seaborn & Regression Plots

- Seaborn is another data visualization libs but it is based on Matlablib
- May generate codes with 5 times less than Matplotlib

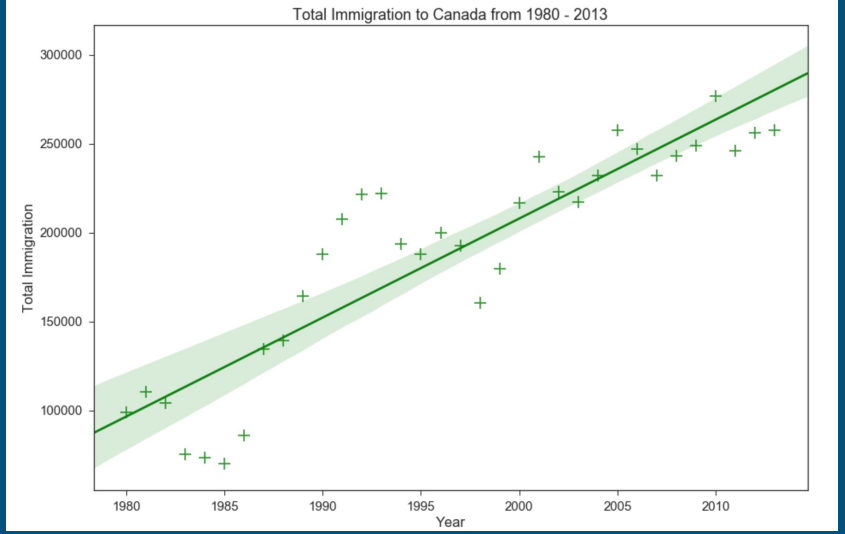
Advanced Visualization Tools - Seaborn & Regression Plots

```
import seaborn as sns
ax = sns.regplot(x='year', y='total', data=df_tot)
```




```
plt.figure(figsize=(15, 10))

sns.set(font_scale=1.5)
sns.set_style('ticks') # change background to white background

ax = sns.regplot(x='year', y='total', data=df_tot, color='green', marker='+', scatter_kws={'s': 200})
ax.set(xlabel='Year', ylabel='Total Immigration')
ax.set_title('Total Immigration to Canada from 1980 - 2013')
```

```
Text(0.5,1,'Total Immigration to Canada from 1980 - 2013')
```



Week 7 - Agenda

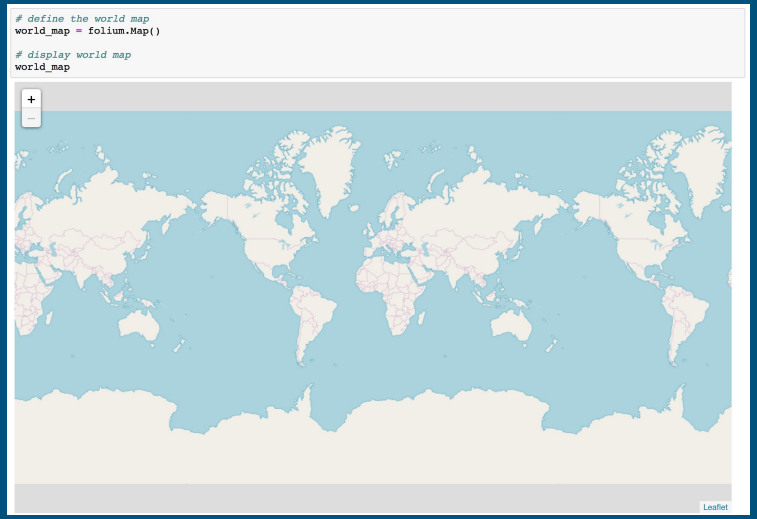
1. Introduction to Visualization Tools
2. Basic Visualization Tools
3. Specialized Visualization Tools
4. Advanced Visualization Tools
5. Creating Maps and Visualizing Geospatial Data

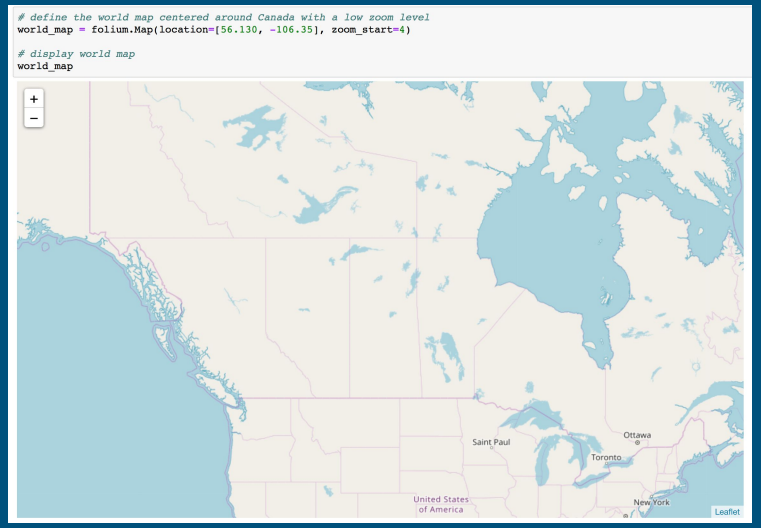
Creating Maps & Visualizing Geospatial Data

1. Visualization Tools
    a. Folium
    b. Maps & Markers
    c. Choropleth Maps

Folium

- Folium is a powerful data visualization library in Python that was built primarily to help people visualize geospatial data.

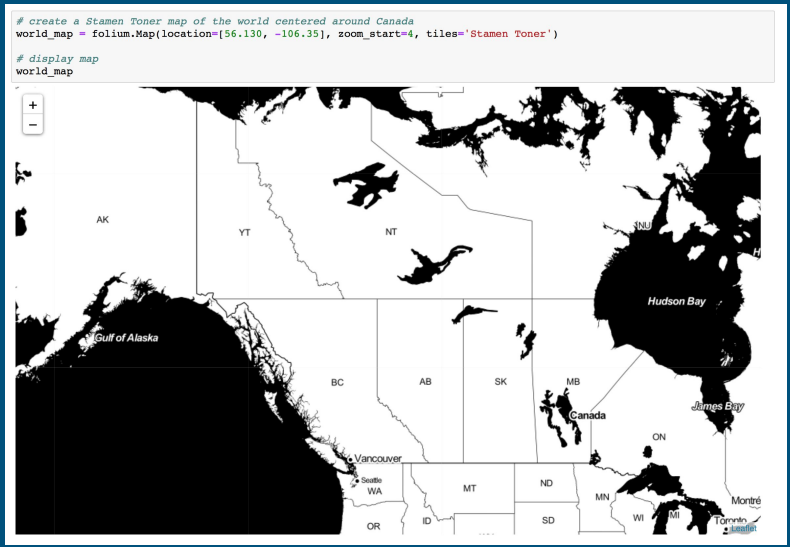




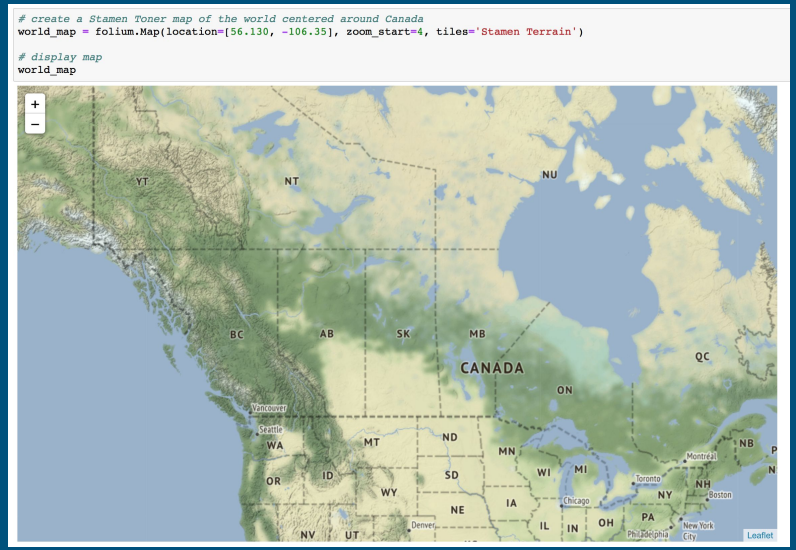
Folium

- Folium is a powerful data visualization library in Python that was built primarily to help people visualize geospatial data.

```
# create a Stamen Toner map of the world centered around Canada
world_map = folium.Map(location=[56.130, -106.35], zoom_start=4, tiles='Stamen Toner')

# display map
world_map
```



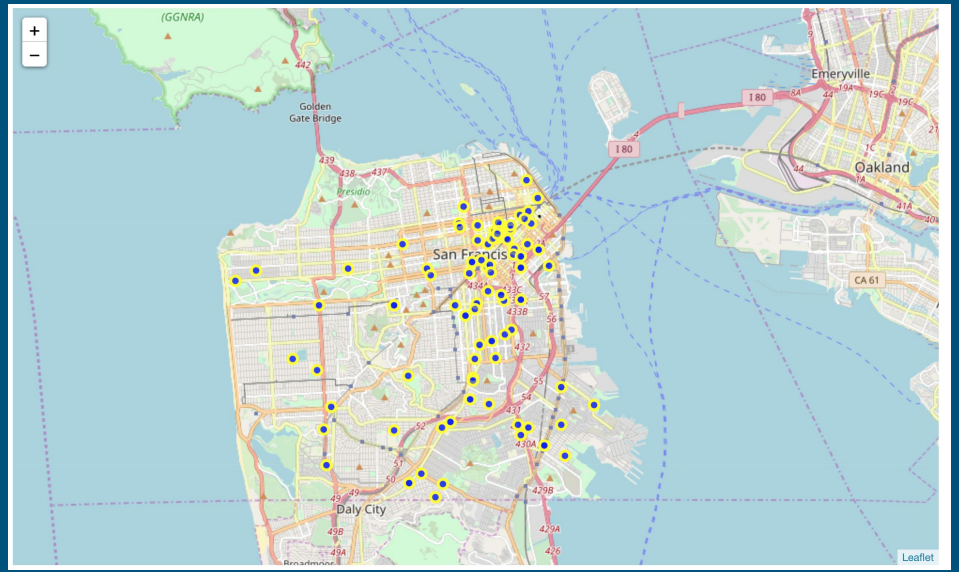

```
# create a Stamen Toner map of the world centered around Canada
world_map = folium.Map(location=[56.130, -106.35], zoom_start=4, tiles='Stamen Terrain')

# display map
world_map
```



Maps & Markers

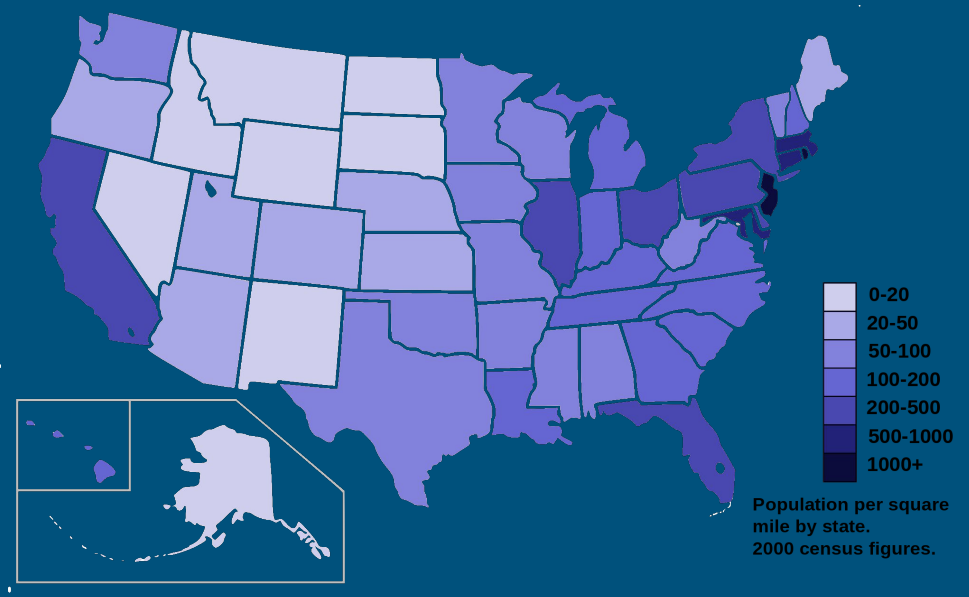
Work with the Folium library and learn how to superimpose markers on top of a map for interesting visualizations.

```
# San Francisco latitude and longitude values
latitude = 37.77
longitude = -122.42

# create map and display it
sanfran_map = folium.Map(location=[latitude, longitude], zoom_start=12)

# display the map of San Francisco
sanfran_map
```

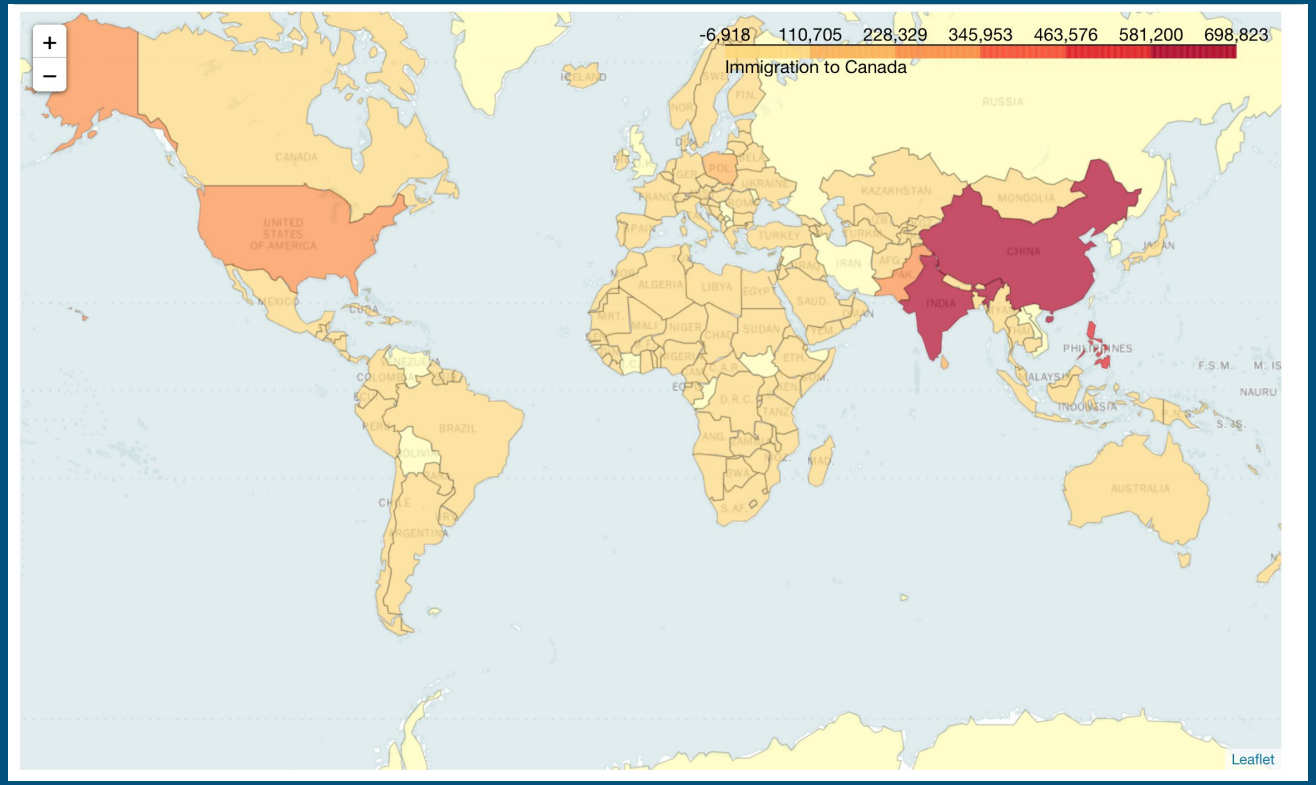


Choropleth Maps

- A Choropleth map is a thematic map in which areas are shaded or patterned in proportion to the measurement of the statistical variable being displayed on the map.



Choropleth Maps



Color Choice

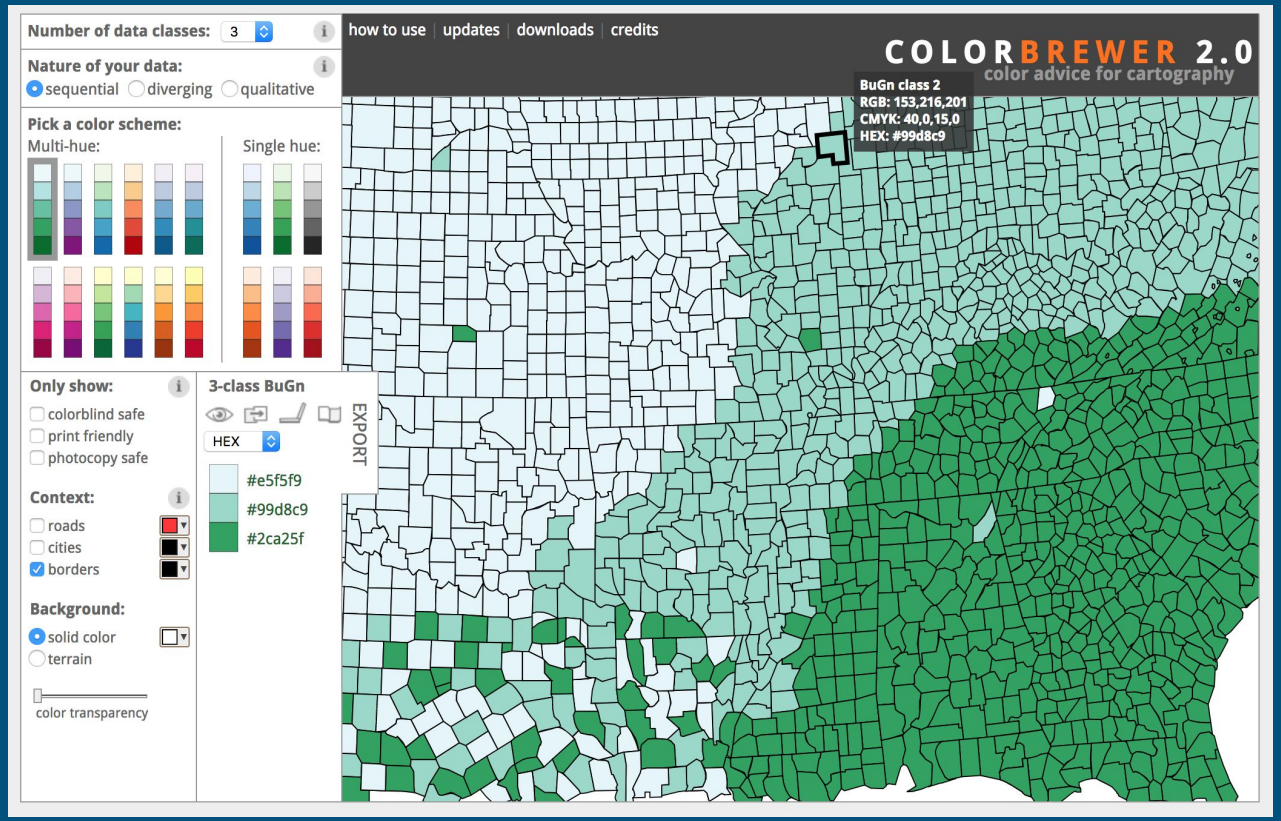
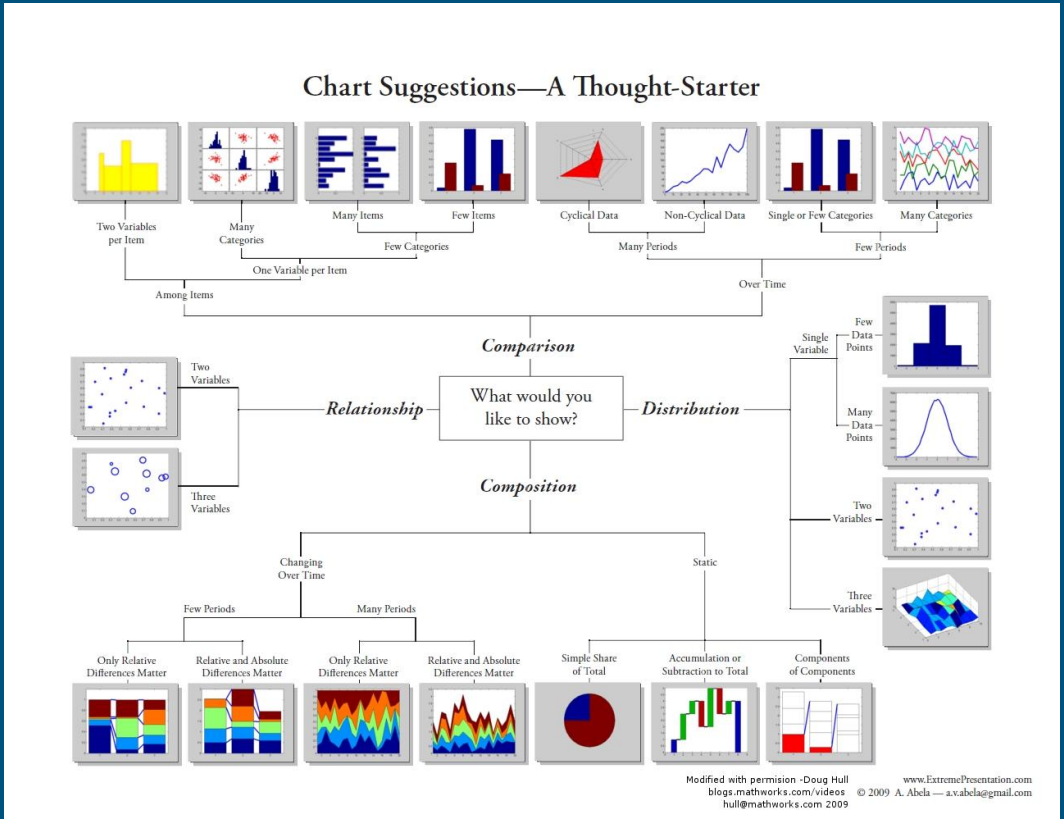


Chart Selection



References

1. https://www.datacamp.com/community/tutorials/seaborn-python-tutorial
2. https://courses.cognitiveclass.ai/courses/course-v1:CognitiveClass+DV0101EN+v1/courseware/407a9f86565c44189740699636b4fb85/12eab34ec218468995e4d06566ef4a32/
3. http://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3
4. https://eazybi.com/blog/data_visualization_and_chart_types/

Thank you