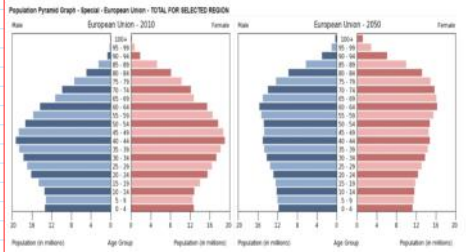
Basic Plots with Matplotlib

Friday, September 25, 2020 9:40 PM

Matplotlib

The mother of all data visualization tools

Graphs and diagrams are meant to tell stories



With a blink of an eye, you can see demographics age (i.e. baby boomers aged 40 in 2010)

Fine tune your graphs to have the best view of reality

- i.e. For Histograms, too few bins will oversimplify reality and won't show you the details. Too many bins will overcomplicate reality and won't show the bigger picture.

Think of the goal first before plotting: i.e You want to visually assess if... H1

Key Notes

Graph Observations

- What's on the x axis and y axis? What are the units? Why are these the units chosen?
- What's the trend and relationship between the variables?

Types of Data Visualization

Line Plot: Plt.plot(x, y) draws a line graph

- Shows a relationship and fills in the gap to tell a story that explains the relationship

Scatterplot: Plt.scatter(x, y) plots the scatter

- Some say this is a more honest representation of the data as you can see the number of data points

Histograph: plt.hist(x)

- Explores a dataset as a first pass visualization - the true power of histograms at work is to tell high level narratives of a single set of data
- Gets an idea about distibution (the height of the bar is the number of bar is in the bin)

Data Visualization

- Many options in plot types and customizations depend on
  1. The data you have
  2. The story you want to tell

Summary

script.py

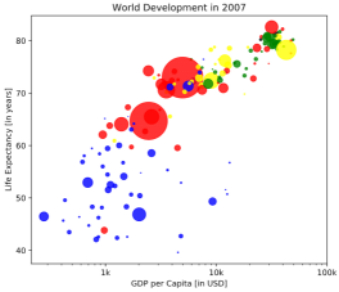
```
# Specify c and alpha inside plt.scatter()
plt.scatter(x = gdp_cap, y = life_exp, s = np.array(pop) * 2, c = col, alpha=0.8)

# Previous customizations
plt.xscale('log')
plt.xlabel('GDP per Capita [in USD]')
plt.ylabel('Life Expectancy [in years]')
plt.title('World Development in 2007')
plt.xticks([1000,10000,100000], ['1k','10k','100k'])

# Show the plot
plt.show()
```

Run Code | Submit Answer

Plots



Previous Plot 1/5 Next Plot

Jupyter Notebook

MatPlotLib

Import Matplotlib

import matplotlib.pyplot as plt

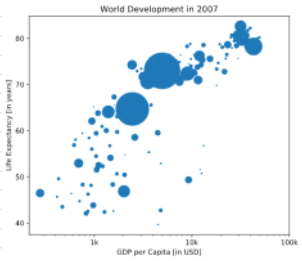
Graphing

Histogram: specifying arguments

```
Help on function hist in module matplotlib.pyplot:
  hist(x, bins=10, range=None, normed=False, weights=None,
  cumulative=False, bottom=None, histtype='bar', align='mid',
  orientation='vertical', rwidth=None, log=False, color=None,
  label=None, stacked=False, hold=None, data=None, **kwargs)
      Plot a histogram.
      Compute and draw the histogram of *x*. The return value is a
      tuple (*n*, *bins*, *patches*) or ([*n0*, *n1*, ...],
      *bins*, [*patches0*, *patches1*,...]) if the input contains
      multiple data.
```

Scatterplot: plt.scatter(gdp_cap, life_exp, s = np_pop)

- Using numpy packages, you can express 3 sets of data on one graph - one variable on the x axis, one on the y axis and one with the size of the bubble!



Checklist

population.py

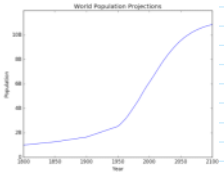
```
import matplotlib.pyplot as plt
year = [1950, 1951, 1952, ..., 2100]
pop = [2.538, 2.57, 2.62, ..., 10.85]

# Add more data
year = [1800, 1850, 1900] + year
pop = [1.0, 1.262, 1.650] + pop

plt.plot(year, pop)

plt.xlabel('Year')
plt.ylabel('Population')
plt.title('World Population Projections')
plt.yticks([0, 2, 4, 6, 8, 10],
           ['0', '2B', '4B', '6B', '8B', '10B'])

plt.show()
```



Requirements

- [ ] Label your Axis
- [ ] Title you Graph
- [ ] Set ticks to express units
- [ ] Add more data (if you have more historical data points)

Optional

- [ ] Set texts on your data points
- [ ] Insert grid lines
- [ ] Set colors on segmented data