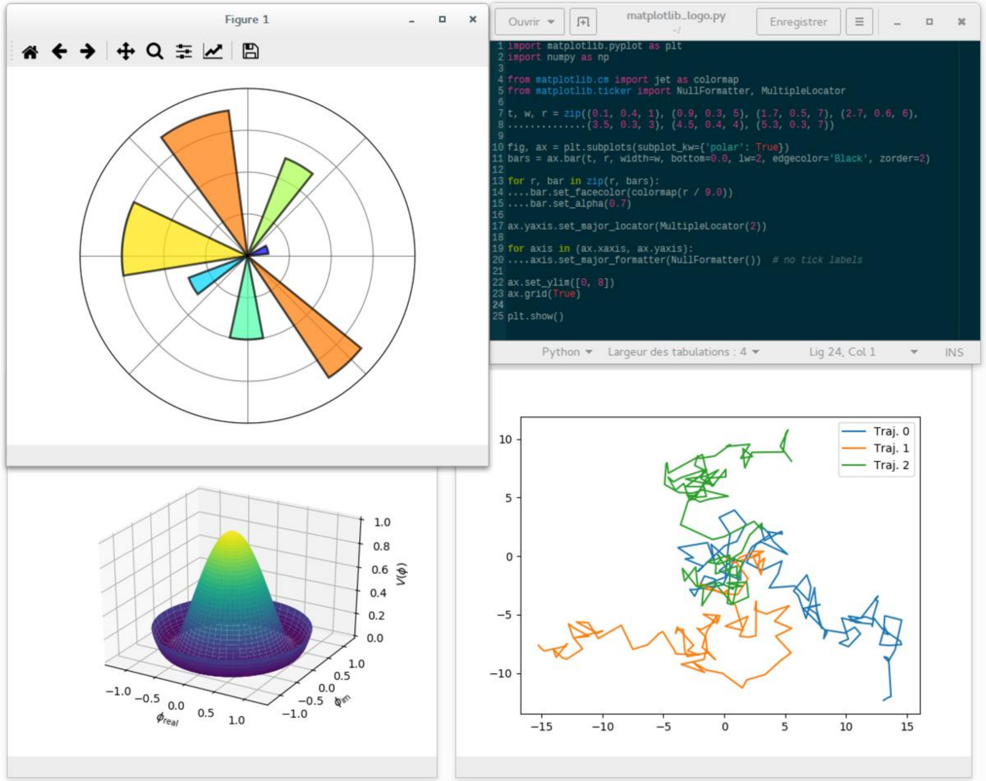


```python
import matplotlib.pyplot as plt
import numpy as np

from matplotlib.cm import jet as colormap
from matplotlib.ticker import NullFormatter, MultipleLocator

t, w, r = zip((0.1, 0.4, 1), (0.9, 0.3, 5), (1.7, 0.5, 7), (2.7, 0.6, 6),
..............(3.5, 0.3, 3), (4.5, 0.4, 4), (5.3, 0.3, 7))

fig, ax = plt.subplots(subplot_kw={'polar': True})
bars = ax.bar(t, r, width=w, bottom=0.0, lw=2, edgecolor='Black', zorder=2)

for r, bar in zip(r, bars):
....bar.set_facecolor(colormap(r / 9.0))
....bar.set_alpha(0.7)

ax.yaxis.set_major_locator(MultipleLocator(2))

for axis in (ax.xaxis, ax.yaxis):
....axis.set_major_formatter(NullFormatter())  # no tick labels

ax.set_ylim([0, 8])
ax.grid(True)

plt.show()
```

Python   Largeur des tabulations : 4   Lig 24, Col 1   INS

Plotting

LECTURE8

Organization of Lecture 8

- Basic Plotting
- Getting to Know Plot Types
- Mastering Embellishments
- Plotting with Pandas

Basic Plotting

- Essential to exploratory or predictive data analysis
- Essential to report writing

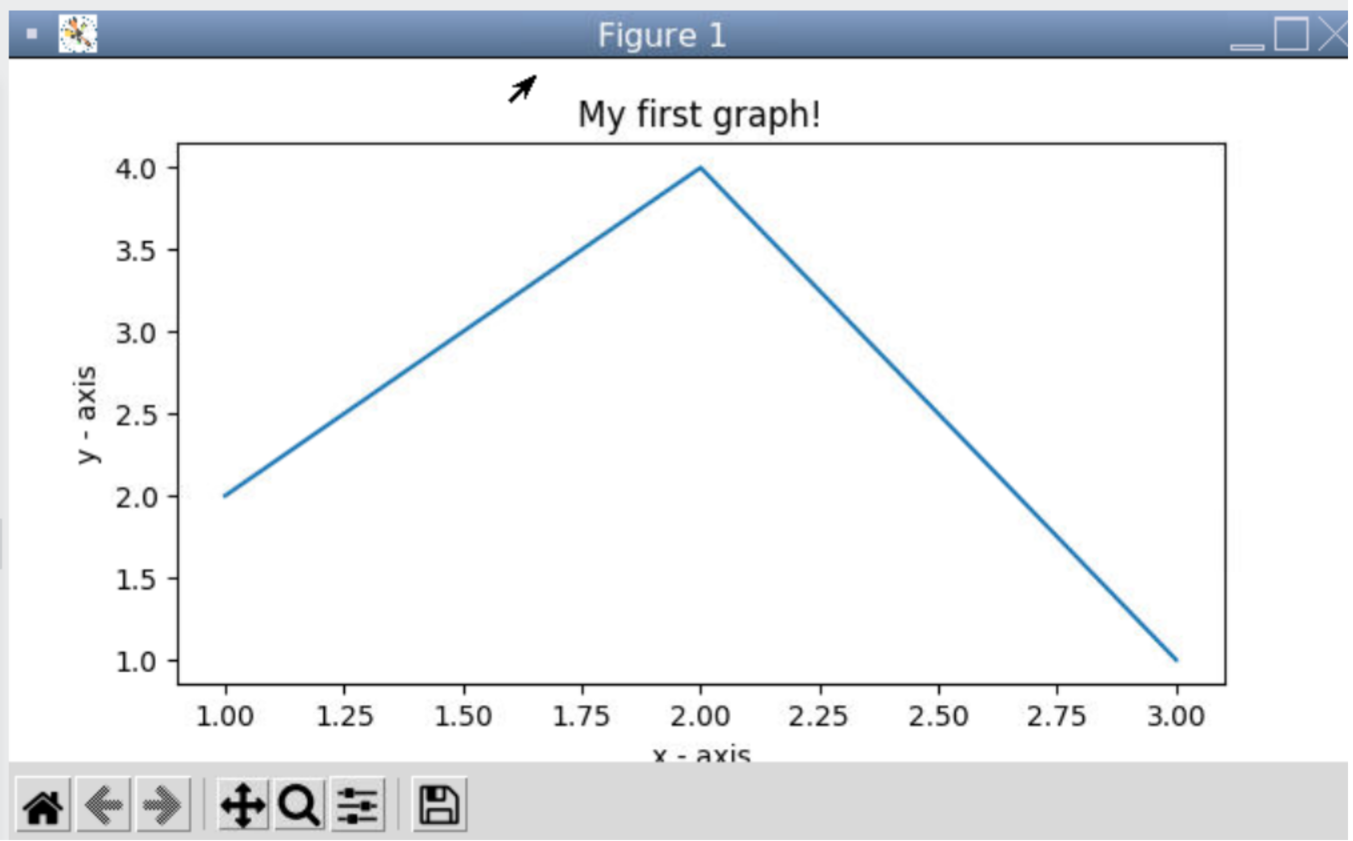
- Three Approaches to programmable plotting
  - Incremental plot: blank canvas and add graphs, axes, labels, legends, etc. / pyplot
  - Monolithic plot: pass all parameters, describing everything / R’s xyplot()
  - Layered plot: what to plot, how to plot, additional features as virtual “layers / matplotlib

Basic Plotting

- numpy and pandas plotting provided by
  - matplotlib (sub-module pyplot)
- pyplot
  - Incremental plotting
  - No single function does all the plotting

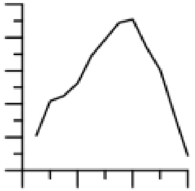
First pyplot Program

```
# importing the required module
import matplotlib.pyplot as plt

# x axis values and y axis values
x = [1,2,3]
y = [2,4,1]

# plotting the points
plt.plot(x, y)

# naming the x axis and y axis
plt.xlabel('x - axis')
plt.ylabel('y - axis')

# giving a title to my graph
plt.title('My first graph!')

# function to show the plot
plt.show()
```

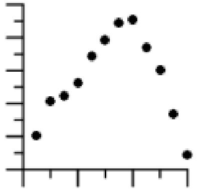


Plot Types

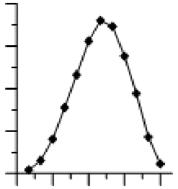
- Line plot: plot()
- Scatter plots: scatter()
- Histogram (vertical or horizontal): hist()
- Pie chart: (pie)
- Bar plot: bar() or barh()
- Box plot: box ()



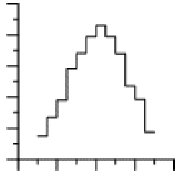
Line Plot



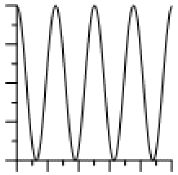
Scatter Plot



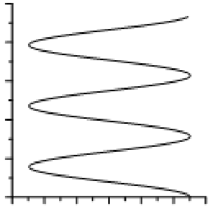
Line/Scatter Plot



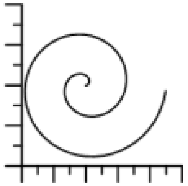
Step Plot



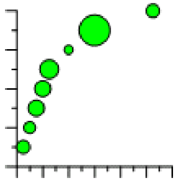
YX Function Plot



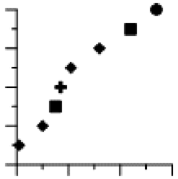
XY Function Plot



Parametric Function Plot

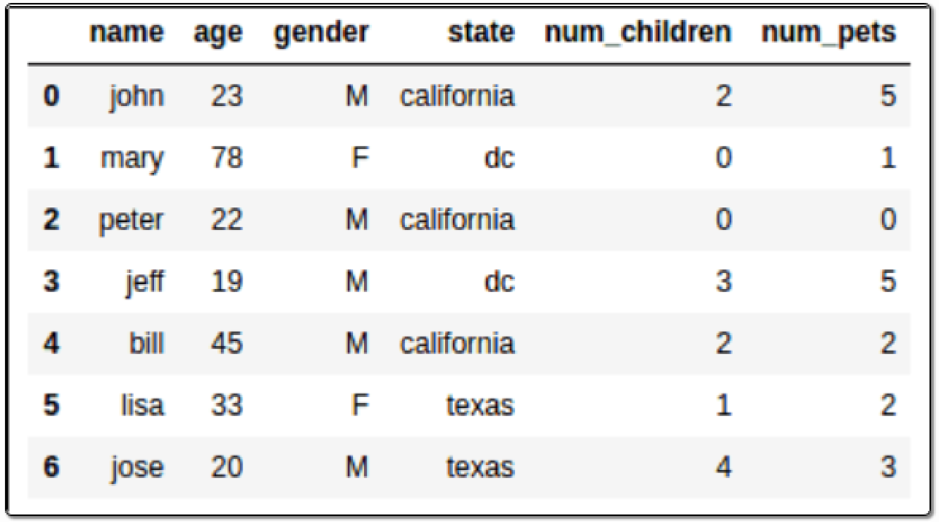


Bubble Plot



Class Plot

Plot Types - Example

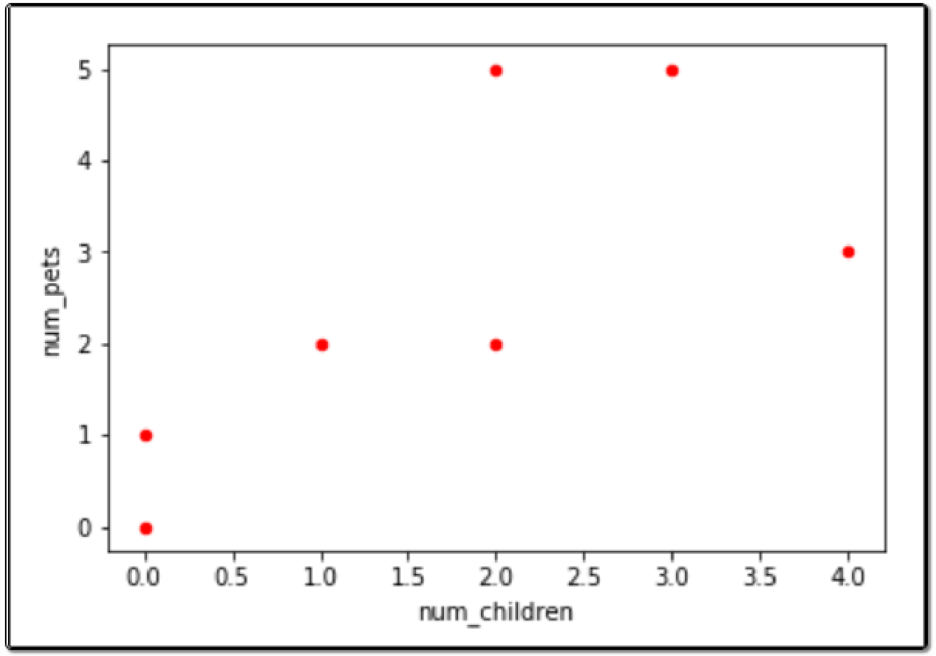


	name	age	gender	state	num_children	num_pets
0	john	23	M	california	2	5
1	mary	78	F	dc	0	1
2	peter	22	M	california	0	0
3	jeff	19	M	dc	3	5
4	bill	45	M	california	2	2
5	lisa	33	F	texas	1	2
6	jose	20	M	texas	4	3

This is what our sample dataset looks like

```
import pandas as pd

df = pd.DataFrame({
    'name':['john','mary','peter','jeff','bill','lisa','jose'],
    'age':[23,78,22,19,45,33,20],
    'gender':['M','F','M','M','M','F','M'],

'state':['california','dc','california','dc','california','texas','texas'],
    'num_children':[2,0,0,3,2,1,4],
    'num_pets':[5,1,0,5,2,2,3]
})
```

Example: Scatter Plot

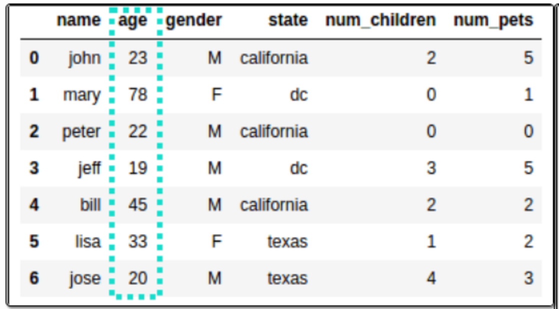


Looks like we have a trend

```
import matplotlib.pyplot as plt
import pandas as pd

# a scatter plot comparing num_children and num_pets
df.plot(kind='scatter',x='num_children',y='num_pets',color='red')
plt.show()
```

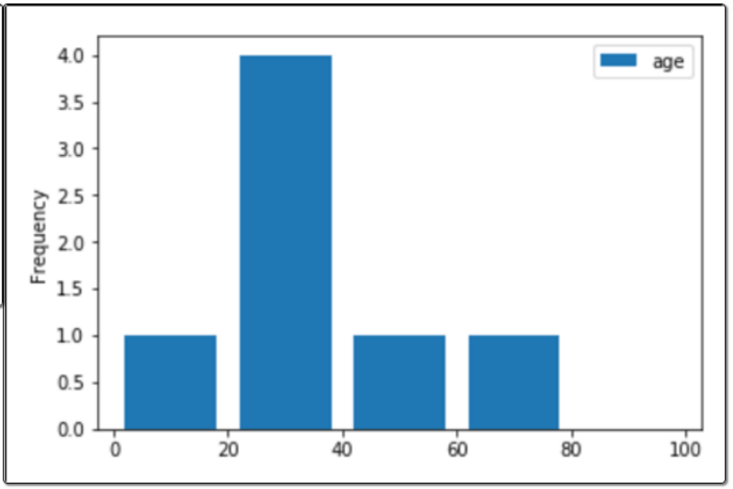
Example: Histogram

```
import matplotlib.pyplot as plt
import pandas as pd

df[['age']].plot(kind='hist',bins=[0,20,40,60,80,100],rwidth=0.8)
plt.show()
```



	name	age	gender	state	num_children	num_pets
0	john	23	M	california	2	5
1	mary	78	F	dc	0	1
2	peter	22	M	california	0	0
3	jeff	19	M	dc	3	5
4	bill	45	M	california	2	2
5	lisa	33	F	texas	1	2
6	jose	20	M	texas	4	3

Source dataframe



The most common age group is between 20 and 40 years old

More Examples

- Dataframe plot-examples with matplotlib pyplot

Embellishments

- With pyplot, you can control lot of aspects of plotting
- You can:
  - Set and change axes scales (“linear” vs “log”) with the `xscale()` and `yscale()` functinos
  - Set and change axes limits with `xlim(xmin,xmax)` and `ylin(ymin, ymax)`
  - Set and change font, graph, and background colors, and font and point sizes and styles
  - Add notes with `annotate()`
  - Add arrows with `arrow()`
  - Add legend with `legend()`

Embellishment Example

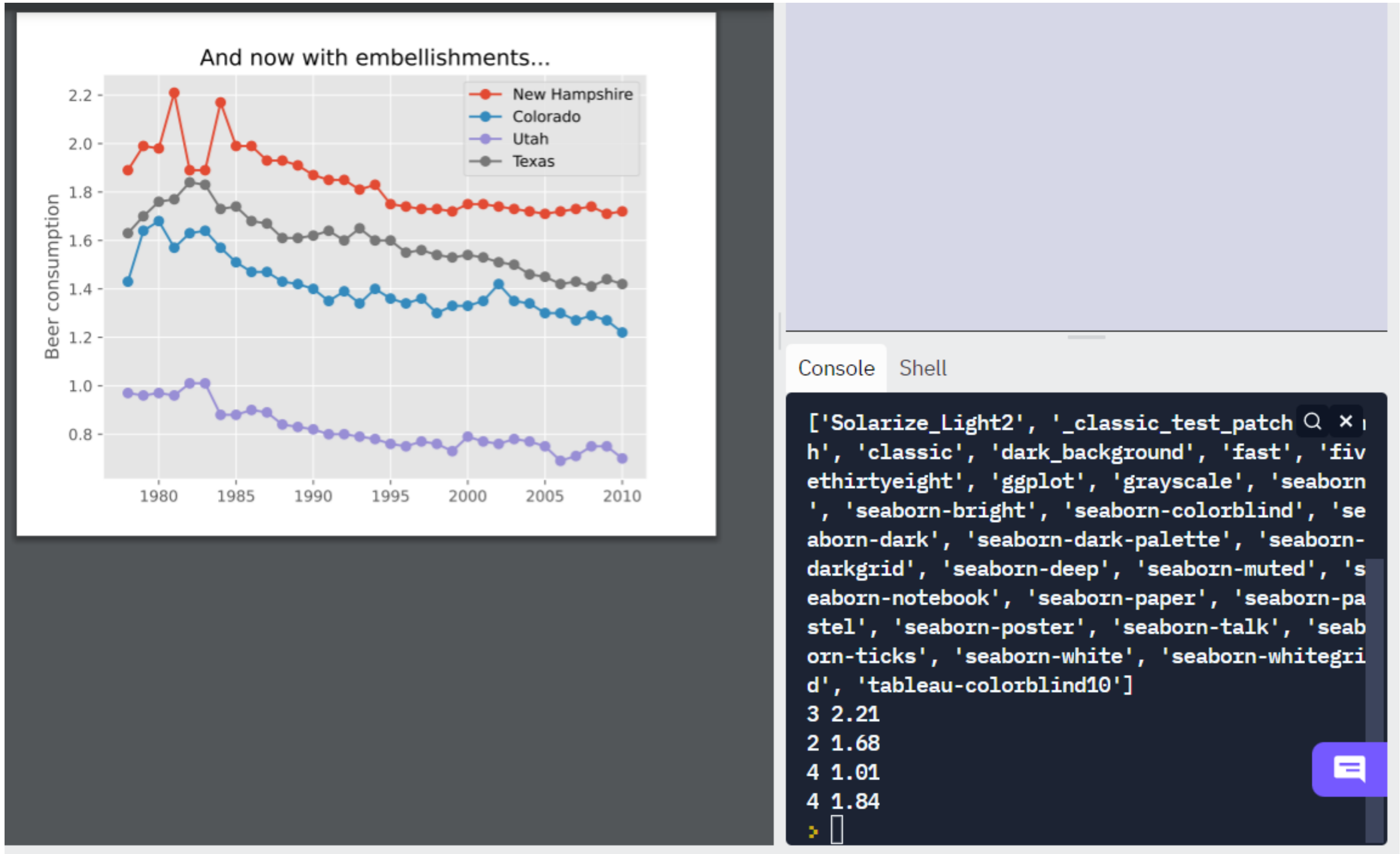
- read_csv() niaaa report
  - has header and is multi-indexed (2); in ascending order of state and year
- Select “beer” for beverage and 4 states (NH, CO, UT, TX) to display
- Select style: ggplot
- Plot the charts
  - For each state, get the data and plot the data for the years
  - Annotate the maximums
  - Add labels and legends
- Save the figure

```
import matplotlib, matplotlib.pyplot as plt
import pandas as pd
print(plt.style.available)

# The NIAAA data sorted in ascending order
of years
alco = pd.read_csv("niaaa-reportv2.csv",
header = 0, index_col=[0,1])

# Select the right data
BEVERAGE = "Beer"
years = alco.index.levels[1]
states = ("New Hampshire", "Colorado",
"Utah", "Texas")

# Select a good-looking style
#plt.xkcd()
matplotlib.style.use("ggplot")
```

https://replit.com/@CSREPLIT/pyplotlegendpy#main.py

```
# Plot the charts
for state in states:
  ydata = alco.loc[state][BEVERAGE]
  plt.plot(years, ydata, "-o")

  print (ydata.argmax(), ydata.max())

  # Add annotations with arrows
plt.annotate(text="Peak",
       xy=(ydata.argmax(), ydata.max()),\
       xytext=(ydata.argmax() + 0.5, \
       ydata.max() + 0.1),\
    arrowprops= dict(facecolor='black', shrink=0.2))

# Add labels and legends
plt.ylabel(BEVERAGE + " consumption")
plt.title("And now with embellishments...")
plt.legend(states)

plt.savefig("embellishedPlot.pdf")
```

pyplot_legend.py output



```
['Solarize_Light2', '_classic_test_patch
h', 'classic', 'dark_background', 'fast', 'fiv
ethirtyeight', 'ggplot', 'grayscale', 'seaborn
', 'seaborn-bright', 'seaborn-colorblind', 'se
aborn-dark', 'seaborn-dark-palette', 'seaborn-
darkgrid', 'seaborn-deep', 'seaborn-muted', 's
eaborn-notebook', 'seaborn-paper', 'seaborn-pa
stel', 'seaborn-poster', 'seaborn-talk', 'seab
orn-ticks', 'seaborn-white', 'seaborn-whitegri
d', 'tableau-colorblind10']
3 2.21
2 1.68
4 1.01
4 1.84
```

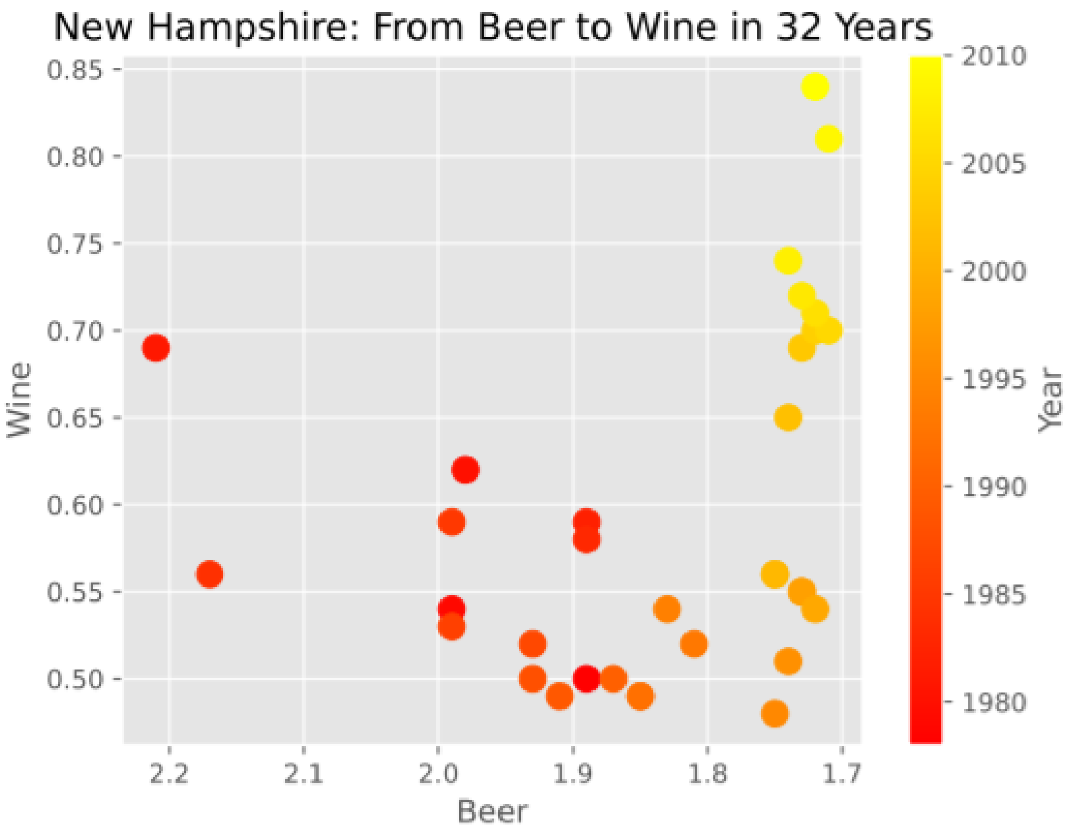
Plotting with Pandas

- `pandas` frames and series supporting plotting through pyplot
- When `plot`() function is called
  - without parameters, it line-plots either the series or all frame columns with labels.
  - With optional parameters `x` and `y`, the function plots column, x against column y
- `pandas` also supports other types of plots with optional parameter kind()
  - All plots allow variety of embellishments, such as dot sizes (option s) and colors (option c)

Scatter Plot Example

- read_csv() niaaa report
  - has header and is multi-indexed (2); in ascending order of state and year
- Select style: ggplot
- Scatter plot the chart for a state
  - Plot the data of wine vs beer consumption over whole time period
  - Color each data point according to observation year
- Add title
- Save the figure

```
import matplotlib, matplotlib.pyplot as plt
import pandas as pd

# The NIAAA data sorted in ascending order of years
alco = pd.read_csv("niaaa-reportv2.csv", header = 0, index_col=[0,1])

# Select a good-locking style
matplotlib.style.use("ggplot")

# Do the scatter plot
STATE = "New Hampshire"
statedata = alco.loc[STATE].reset_index()
scatter = statedata.plot.scatter("Beer", "Wine", c="Year", \
                                 s=100, cmap=plt.cm.autumn)

#REVERSE x-axis
ax = scatter.axes
ax.invert_xaxis()

plt.title("%s: From Beer to Wine in 32 Years" % STATE)
plt.savefig("scatter-plot.pdf")
```



https://replit.com/@CSREPLIT/scatter-plot#main.py

Plotting Scatter Matrices

- pandas has a submodule `pandas.tools.plotting`
- ONE of the tools is scatter matrices
  - Excellent exploratory instrument
  - Displays histograms for each column in the main diagonal and two-variable scatter plots for each combination of two columns

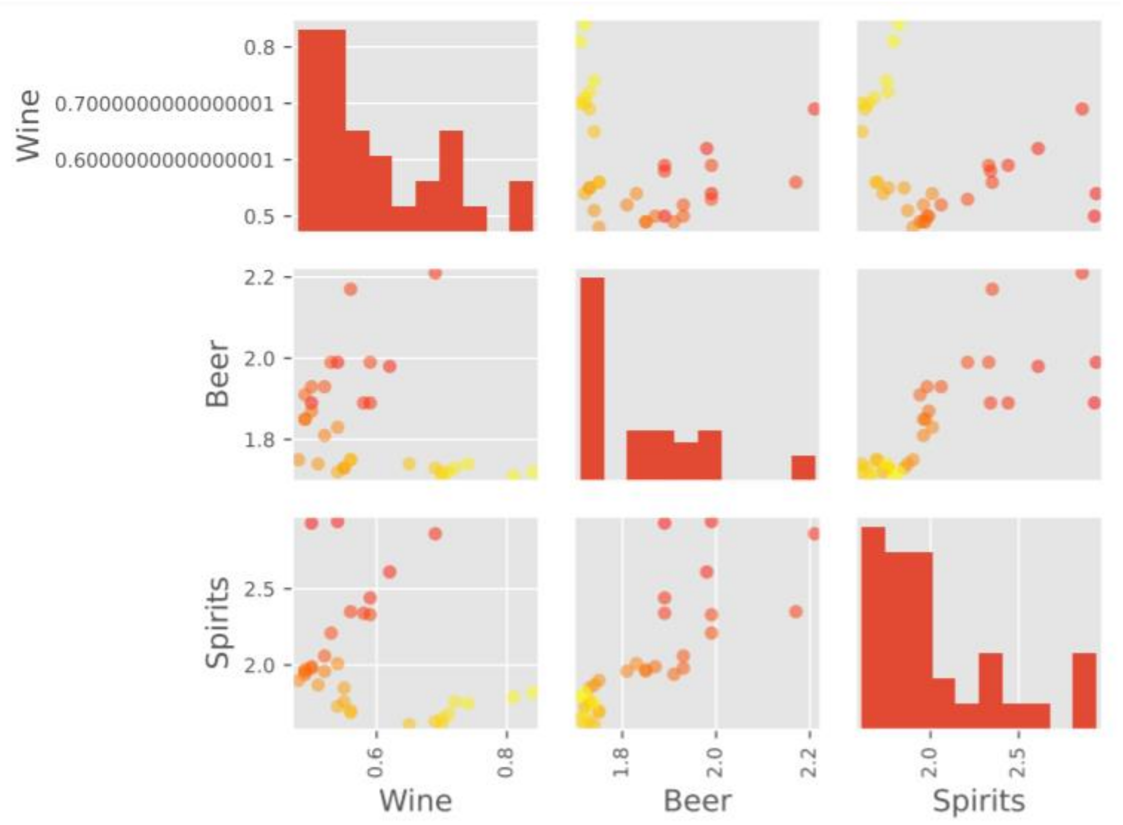
Scatter Matrix Plot Example

- Search for pandas.tools.plotting and add package in left frame
- read_csv() niaaa report
  - has header and is multi-indexed (2); in ascending order of state and year
- Select style: ggplot
- Scatter plot the scatter matrix
  - Choose a state
  - Plot the scatter matrix
  - Choose layout
- Save the figure

```
from pandas.plotting import scatter_matrix   #different from text
import matplotlib, matplotlib.pyplot as plt
import pandas as pd

# The NIAAA data sorted in ascending order of years
alco = pd.read_csv("niaaa-reportv2.csv", header = 0, index_col=[0,1])

# Select a good-locking style
matplotlib.style.use("ggplot")

# Plot the scatter matrix
STATE = "New Hampshire"
statedata = alco.loc[STATE].reset_index()
scatter_matrix(statedata[["Wine", "Beer", "Spirits"]],
s=120, c=statedata["Year"], cmap=plt.cm.autumn)

plt.tight_layout()
plt.savefig("scatter-matrix.pdf")
```

https://replit.com/@CSREPLIT/Scatter-Matrics#main.py

Scatter Matrix Output


Wine
Beer
Spirits
Wine
Beer
Spirits