Mastering Python
الدرس #2_7
MatPlotLib الرسم البياني ورسم الخرائط

By:

Hussam Hourani

V1.0 - NOV 2019

Agenda

- What is MatPlotLib?
- MatPlotLib Sample Plotting
- line style / marker and Colors
- Advanced Plotting
- 3D Plotting

What is MatPlotLib?

Matplotlib is a Python plotting library

Matplotlib is a large and sophisticated graphics package written in object Oriented style

for plotting several variety of graphs, starting from histograms to line plots to scatter plots,…

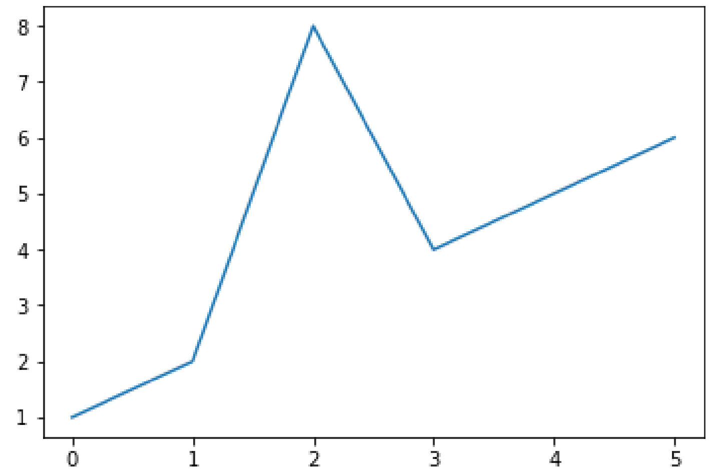
And many more

https://matplotlib.org/

MatPlotLib

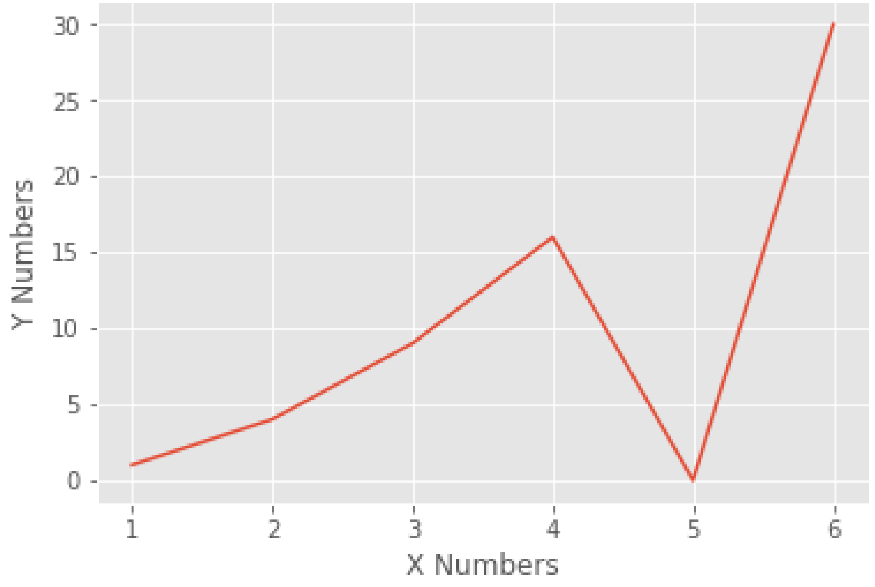
```
import matplotlib.pyplot as plt

f=[1, 2, 8, 4,5,6]

plt.plot(f)
plt.show()
```

Output



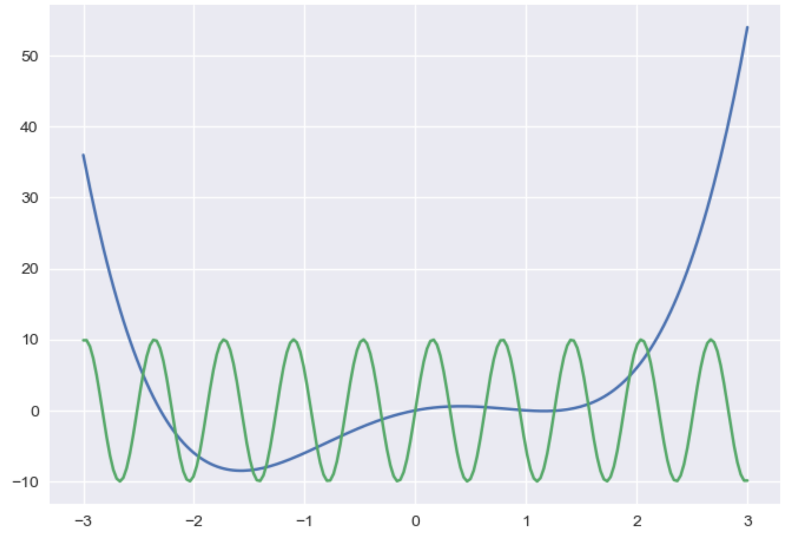
```
import matplotlib.pyplot as plt
plt.style.use('ggplot')

x=[1, 2, 3, 4,5,6]
y=[1, 4, 9, 16,0,30]

plt.plot(x,y )
plt.ylabel('Y Numbers')
plt.xlabel('X Numbers')
plt.show()
```

Output

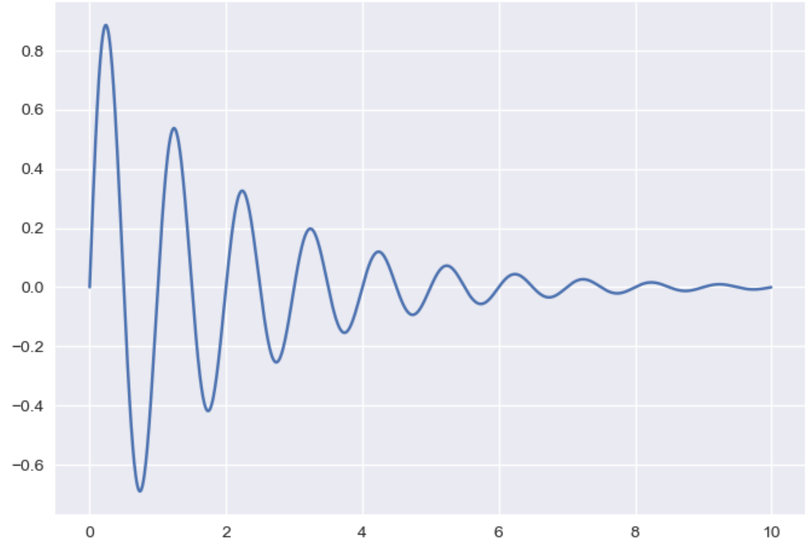


MatPlotLib

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

def p1(x): return x**4 - 4*x**2 + 3*x
def p2(x): return np.sin(10*x) * 10

X = np.linspace(-3, 3, 200)

plt.plot( X,p1(X), X,p2(X))
plt.show()
```



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

x = np.arange(0, 10, 0.005)
y = np.exp(-x/2.) * np.sin(2*np.pi*x)

plt.plot(x, y)
plt.xlim(0, 10)
plt.ylim(-1, 1)
plt.show()
```



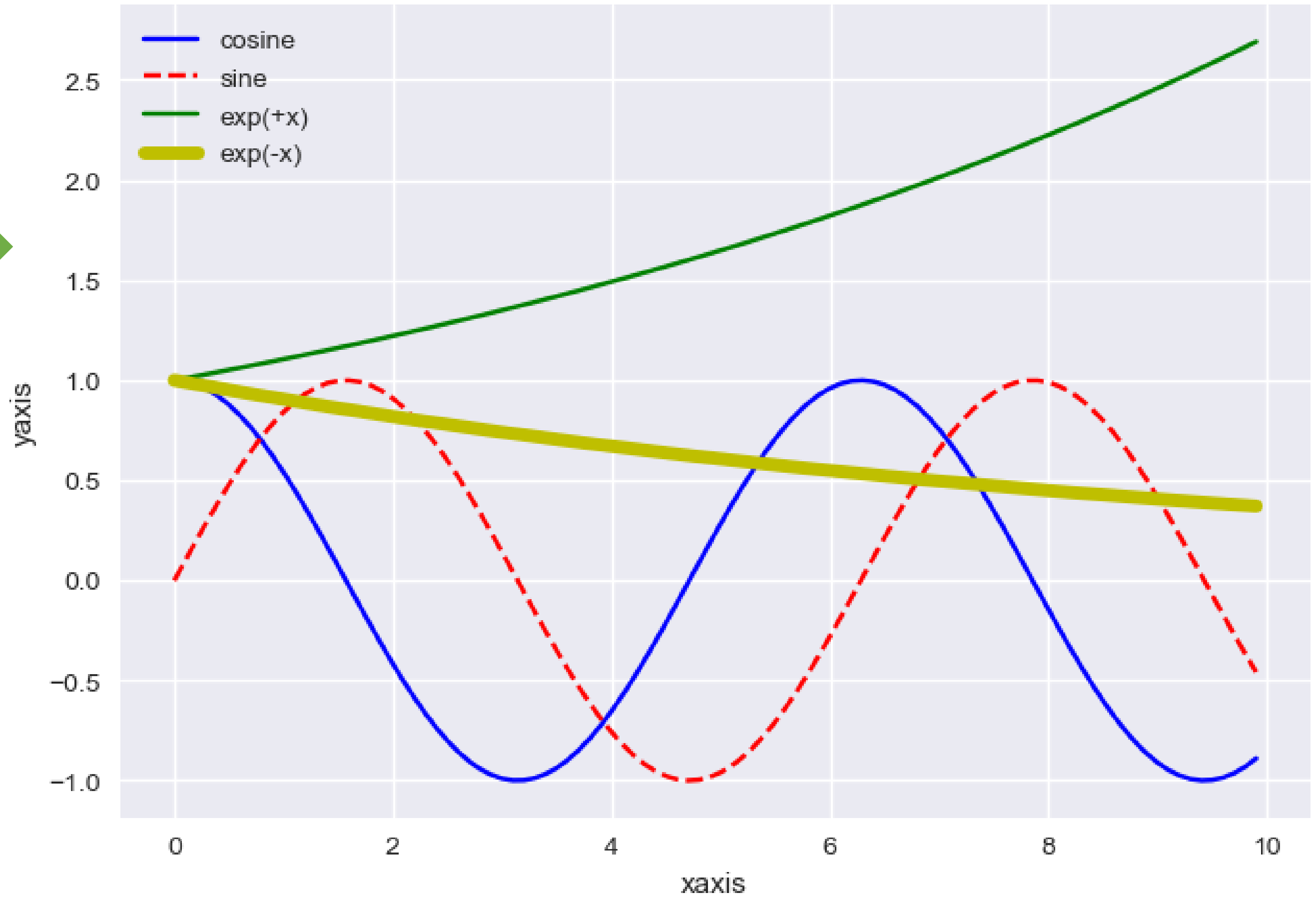
line style / marker and Colors

Character	description
'-'	solid line style
'--'	dashed line style
'-.'	dash-dot line style
':'	dotted line style
'.'	point marker
','	pixel marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
'1'	tri_down marker
'2'	tri_up marker
'3'	tri_left marker
'4'	tri_right marker
's'	square marker
'p'	pentagon marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'D'	diamond marker
'd'	thin_diamond marker
'|'	vline marker
'_'	hline marker

character	color
‘b’	blue
‘g’	green
‘r’	red
‘c’	cyan
‘m’	magenta
‘y’	yellow
‘k’	black
‘w’	white

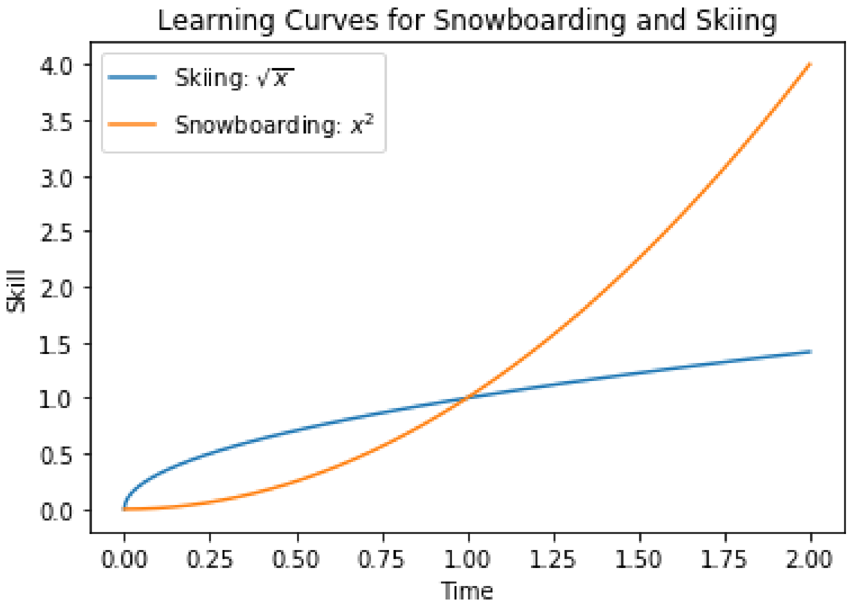
MatPlotLib

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

x=np.arange(0.,10,0.1)
a=np.cos(x)
b=np.sin(x)
c=np.exp(x/10)
d=np.exp(-x/10)

plt.plot(x,a,'b-',label='cosine')
plt.plot(x,b,'r--',label='sine')
plt.plot(x,c,'g-',label='exp(+x)')
plt.plot(x,d,'y-
',linewidth=5,label='exp(-x)')

plt.legend(loc='upperleft')
plt.xlabel('xaxis')
plt.ylabel('yaxis')
plt.show()
```

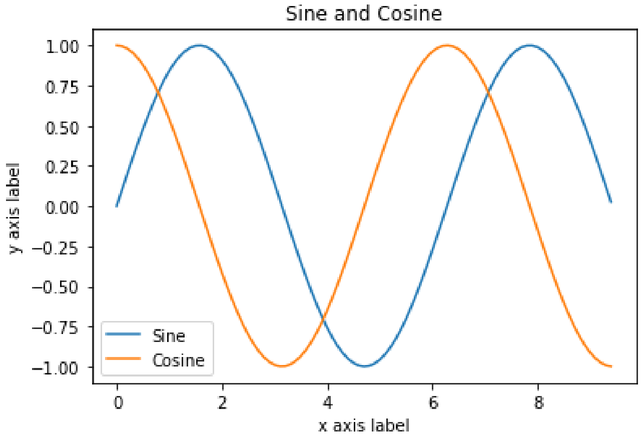


MatPlotLib

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(0,2, 1000)
plt.figure()

plt.plot(x, np.sqrt(x), label = r"Skiing: $\sqrt{x}$")
plt.plot(x, x**2, label = r"Snowboarding: $x^2$")

plt.title("Learning Curves for Snowboarding and Skiing")
plt.xlabel("Time") ; plt.ylabel("Skill")
plt.legend(loc='upper left')
plt.show
```



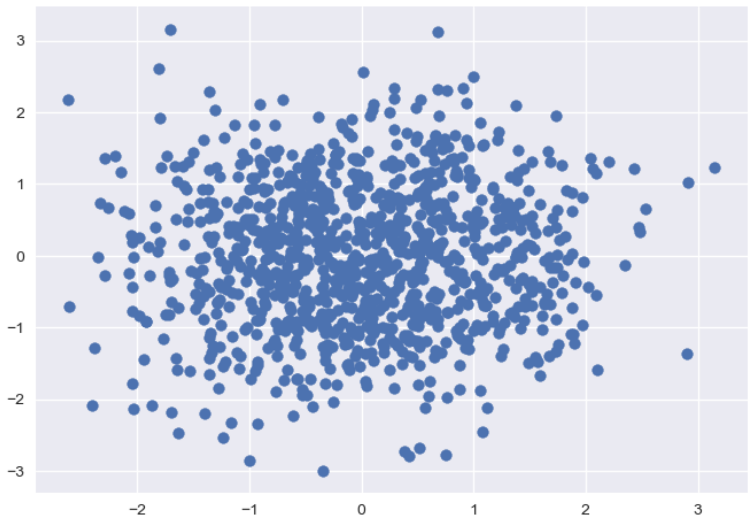
```
import numpy as np
import matplotlib.pyplot as plt
x = np.arange(0, 3 * np.pi, 0.1)
y_sin = np.sin(x);y_cos = np.cos(x)
# Plot the points using matplotlib
plt.plot(x, y_sin); plt.plot(x, y_cos)
plt.xlabel('x axis label'); plt.ylabel('y axis label')
plt.title('Sine and Cosine');plt.legend(['Sine', 'Cosine'])
plt.show()
```



MatPlotLib

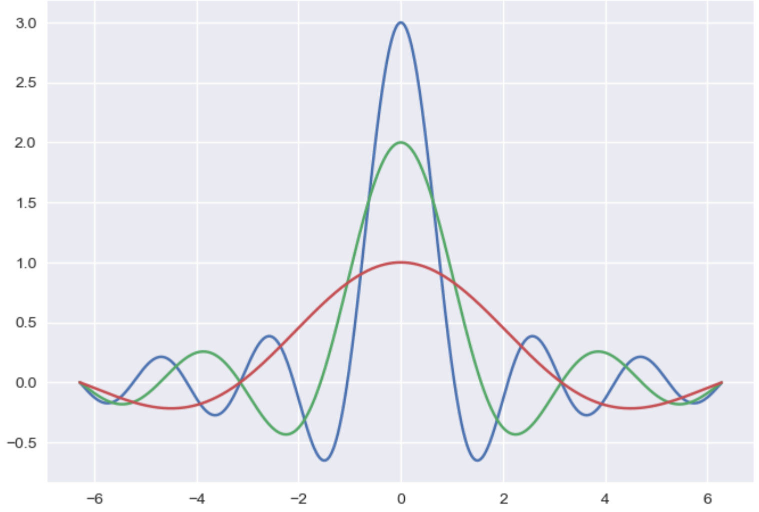
```
import numpy as np
import matplotlib.pyplot as plt

n = 1024
X = np.random.normal(0,1,n)
Y = np.random.normal(0,1,n)

plt.scatter(X,Y)
plt.show()
```

Output



```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(-2*np.pi,2*np.pi,0.01)
y = np.sin(3*x)/x
y2 = np.sin(2*x)/x
y3 = np.sin(x)/x
plt.plot(x,y)
plt.plot(x,y2)
plt.plot(x,y3)
```

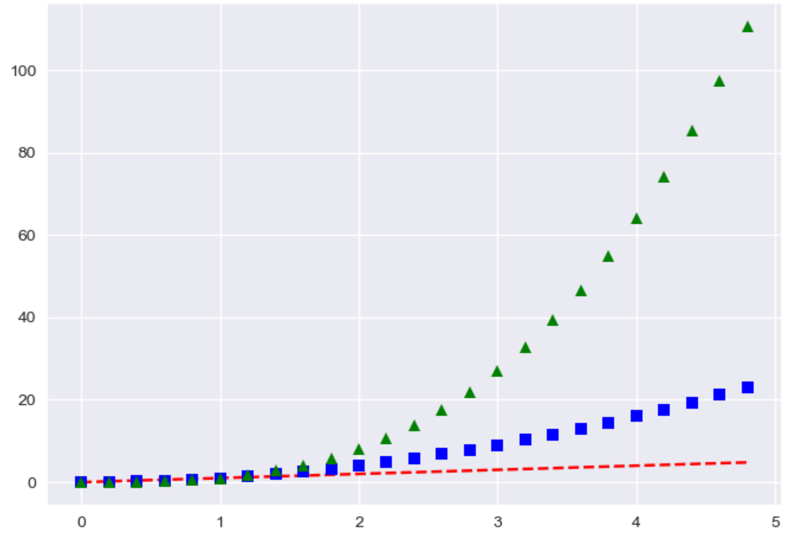
Output



MatPlotLib

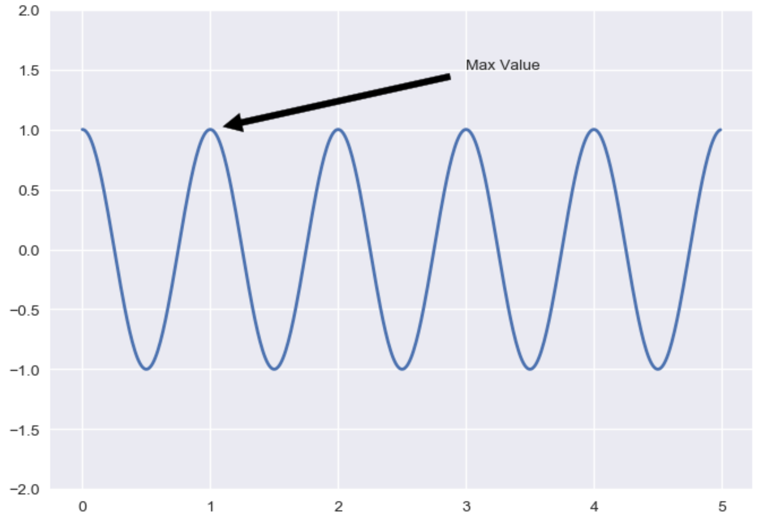
```
import numpy as np
import matplotlib.pyplot as plt
t = np.arange(0., 5., 0.2)
# red dashes, blue squares and green triangles
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3,
'g^')
plt.show()
```

Output



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

t = np.arange(0.0, 5.0, 0.01)
s = np.cos(2*np.pi*t)
plt.plot(t, s, lw=2)   #lw : Line width

plt.annotate('Max Value', xy=(2, 1), xytext=(3,
1.5),arrowprops=dict(facecolor='black', shrink=0.05),)

plt.ylim(-2,2)
plt.show()
```

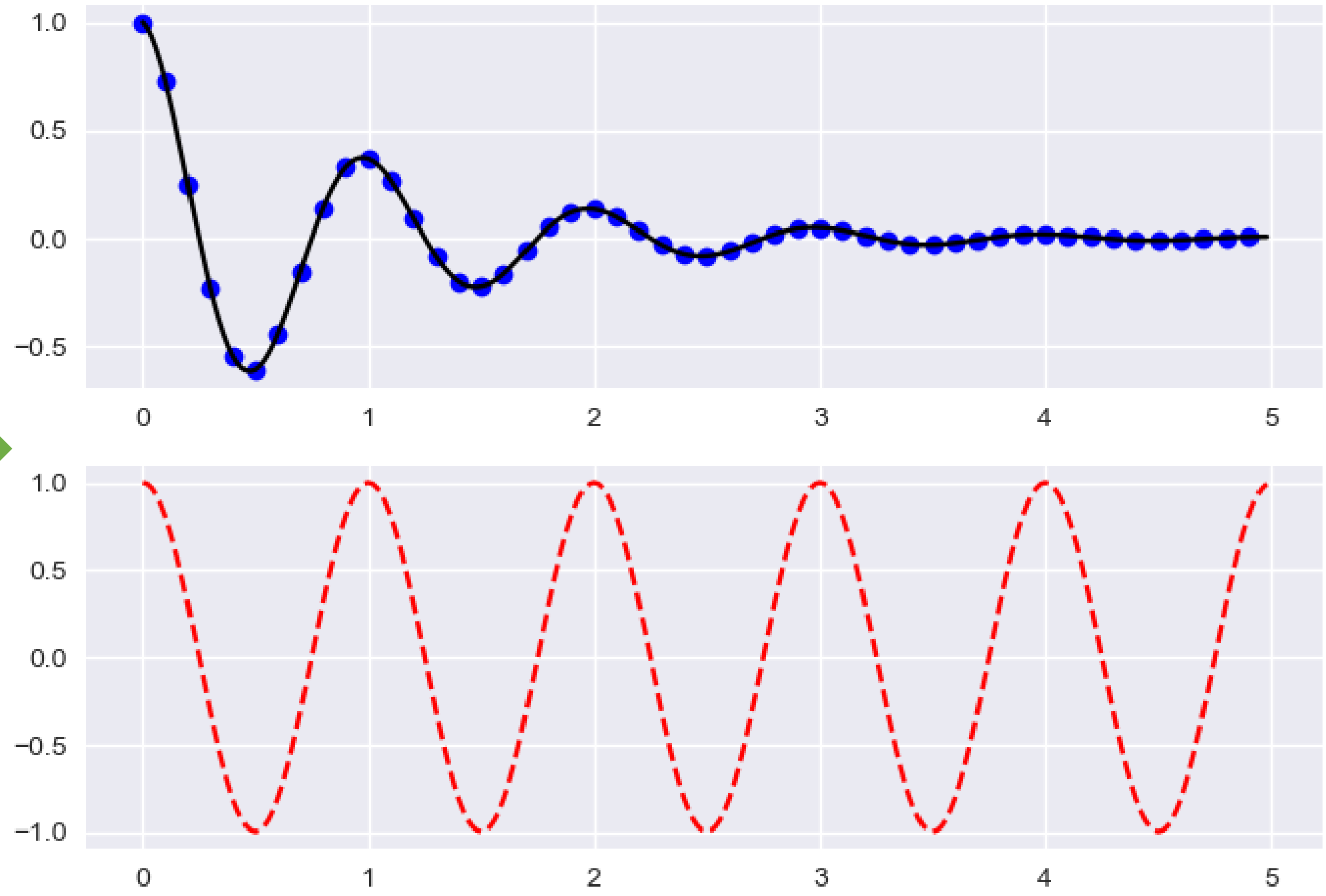
Output



Plotting Two Subplots

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

def f(t):
    return np.exp(-t) * np.cos(2*np.pi*t)

t1 = np.arange(0.0, 5.0, 0.1)
t2 = np.arange(0.0, 5.0, 0.02)

plt.figure(1)

plt.subplot(211)
plt.plot(t1, f(t1), 'bo', t2, f(t2), 'k')

plt.subplot(212)
plt.plot(t2, np.cos(2*np.pi*t2), 'r--')

plt.show()
```

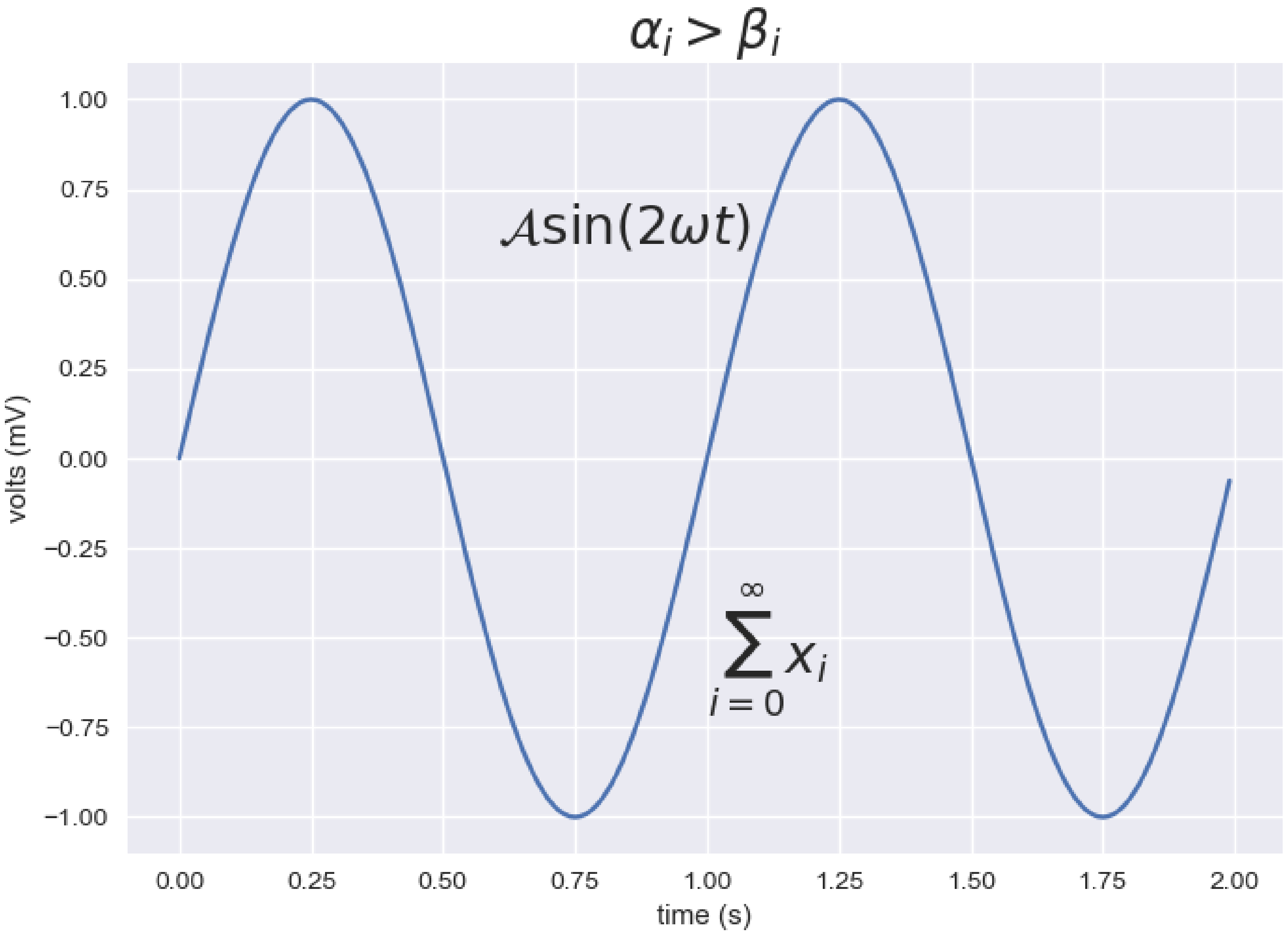
Output



Plotting Two Subplots

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

t = np.arange(0.0, 2.0, 0.01)
s = np.sin(2*np.pi*t)

plt.plot(t,s)
plt.title(r'$\alpha_i > \beta_i$', fontsize=20)
plt.text(1, -0.6, r'$\sum_{i=0}^\infty x_i$',
fontsize=20)
plt.text(0.6, 0.6, r'$\mathcal{A}\mathrm{sin}(2
\omega t)$',fontsize=20)

plt.xlabel('time (s)')
plt.ylabel('volts (mV)')
plt.show()
```

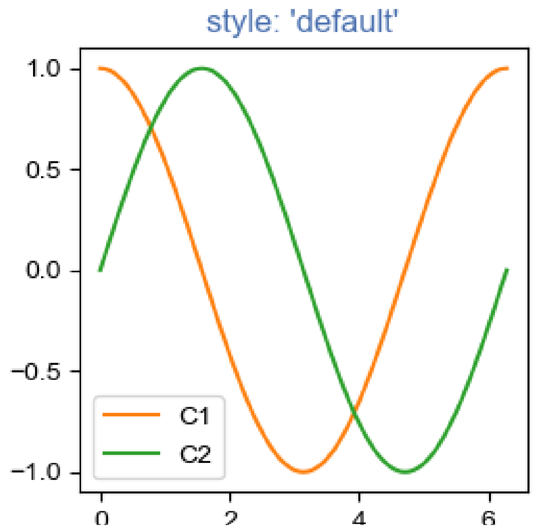
Output

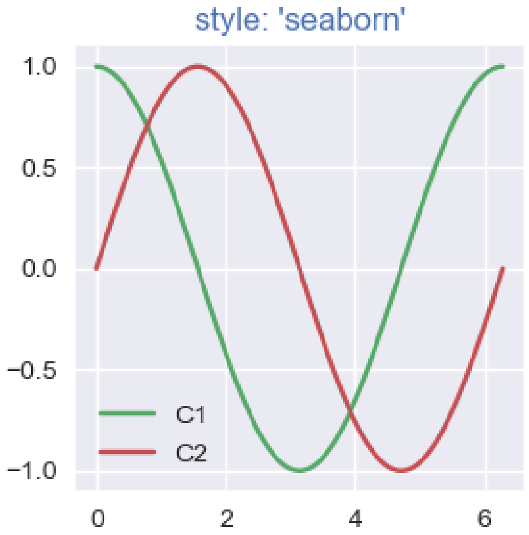


MatPlotLib

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

th = np.linspace(0, 2*np.pi, 128)


def demo(sty):
    mpl.style.use(sty)
    fig, ax = plt.subplots(figsize=(3, 3))

    ax.set_title('style: {!r}'.format(sty), color='C0')

    ax.plot(th, np.cos(th), 'C1', label='C1')
    ax.plot(th, np.sin(th), 'C2', label='C2')
    ax.legend()

demo('default')
demo('seaborn')
```

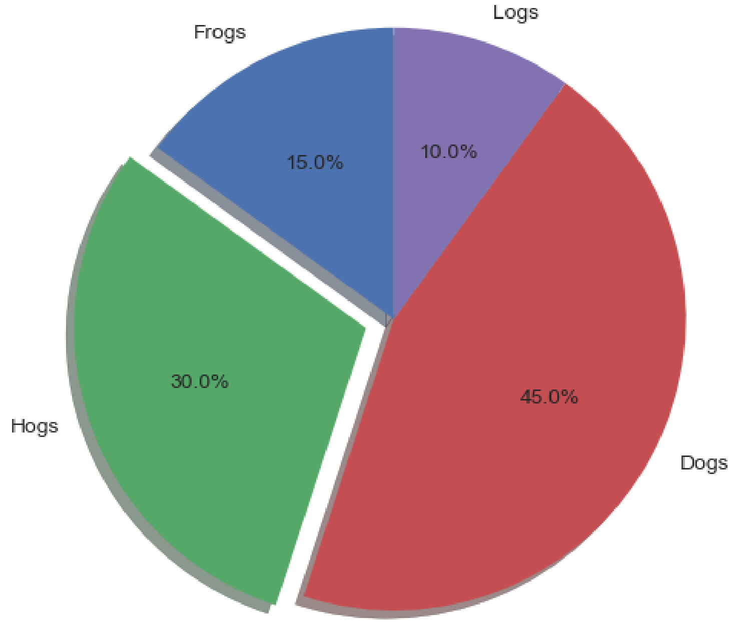
Output



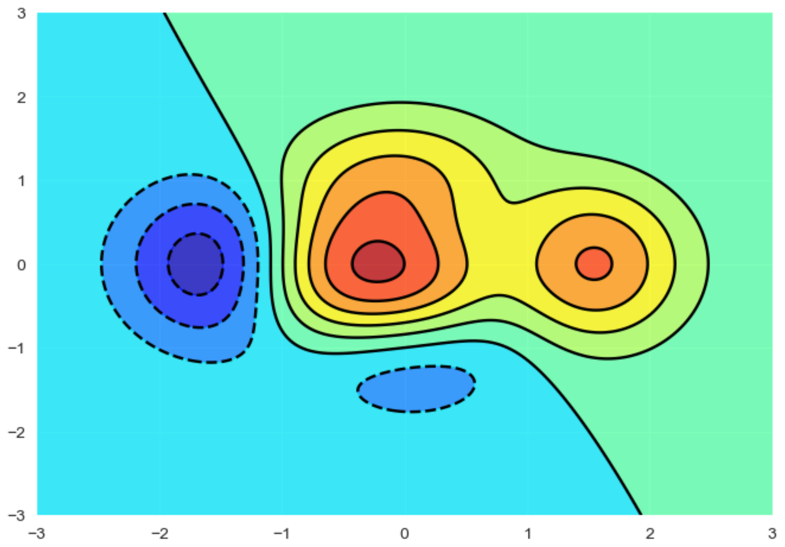


MatPlotLib

```
import matplotlib.pyplot as plt
labels = 'Frogs', 'Hogs', 'Dogs', 'Logs'
sizes = [15, 30, 45, 10]
explode = (0, 0.1, 0, 0)
fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels,
autopct='%1.1f%%',shadow=True, startangle=90)
ax1.axis('equal')
plt.show()
```



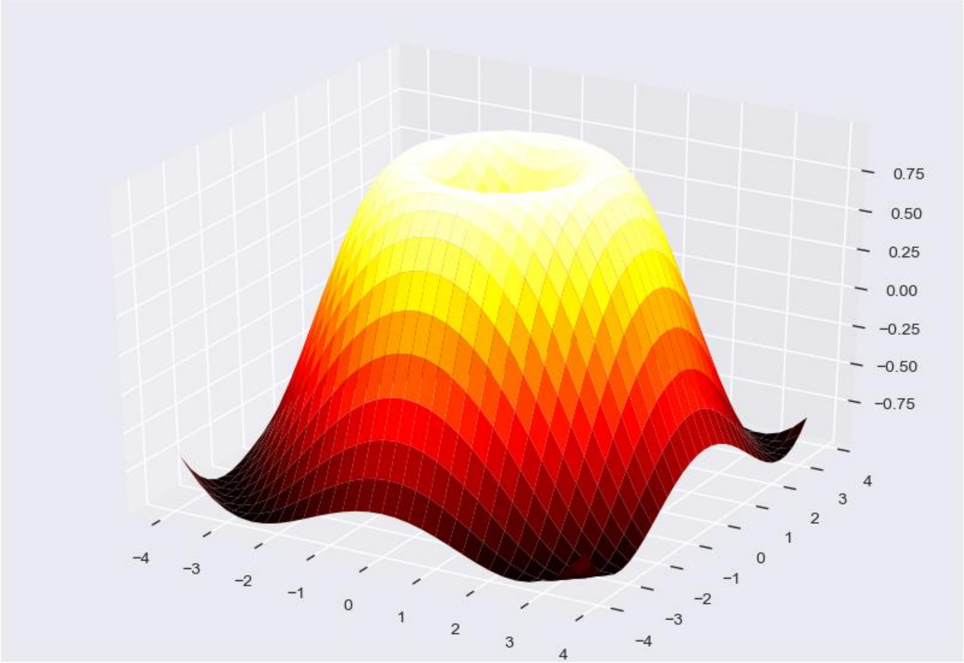
```
import numpy as np
import matplotlib.pyplot as plt
def f(x,y): return (1-x/2+x**5+y**3)*np.exp(-x**2-y**2)
n = 256
x = np.linspace(-3,3,n)
y = np.linspace(-3,3,n)
X,Y = np.meshgrid(x,y)

plt.contourf(X, Y, f(X,Y), 8, alpha=.75, cmap='jet')
plt.contour(X, Y, f(X,Y), 8, colors='black',
linewidth=.5)
plt.show()
```



3D Plotting

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = Axes3D(fig)
X = np.arange(-4, 4, 0.25)
Y = np.arange(-4, 4, 0.25)
# meshgrid : Return coordinate matrices from coordinate
vectors.
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X**2 + Y**2)
Z = np.sin(R)

ax.plot_surface(X, Y, Z, cmap='hot')

plt.show()
```

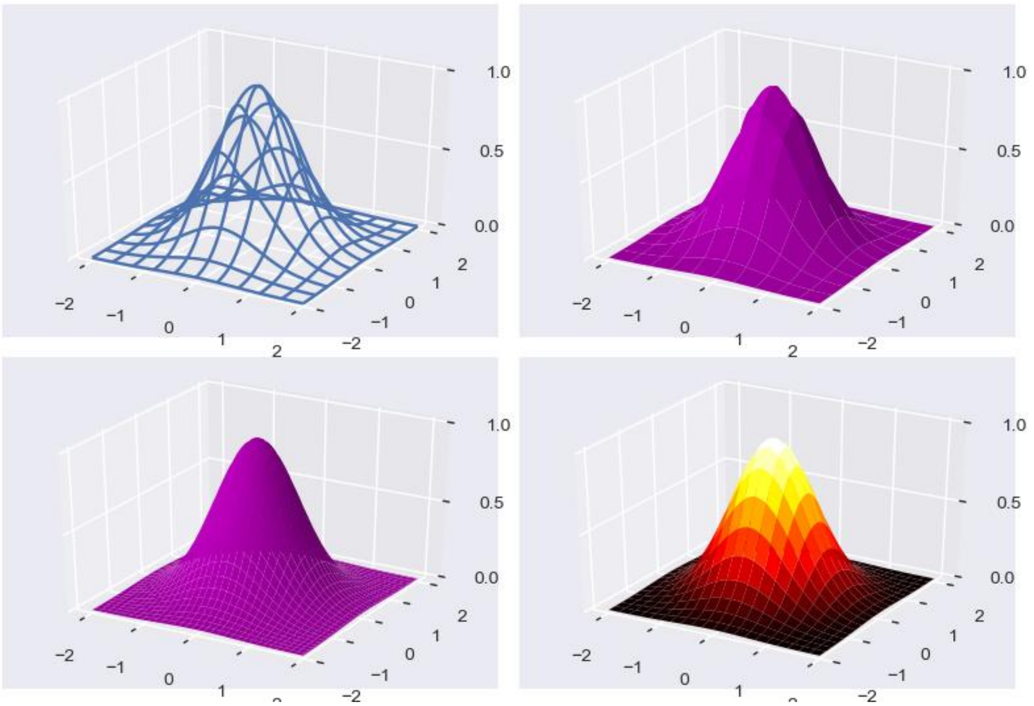
Output



Surface Plotting

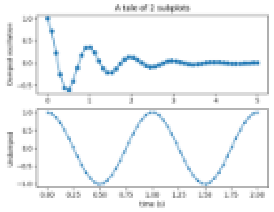
```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

x = np.linspace(-2, 2, 400)
y = x.copy()
X, Y = np.meshgrid(x, y)
Z = np.exp(-(X**2 + Y**2))

fig, ax = plt.subplots(nrows=2, ncols=2,
subplot_kw={'projection': '3d'})

ax[0,0].plot_wireframe(X, Y, Z, rstride=40, cstride=40)
ax[0,1].plot_surface(X, Y, Z, rstride=40, cstride=40, color='m')
ax[1,0].plot_surface(X, Y, Z, rstride=12, cstride=12, color='m')
ax[1,1].plot_surface(X, Y, Z, rstride=20, cstride=20, cmap='hot')

fig.tight_layout()
plt.show()
```

Output

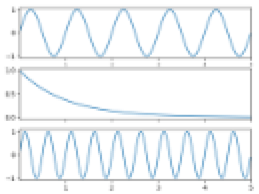


https://scipython.com/book/chapter-7-matplotlib/examples/simple-surface-plots/

Other Plotting Samples



Multiple subplots



Shared Axis Demo

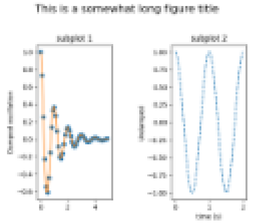
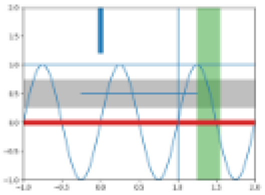
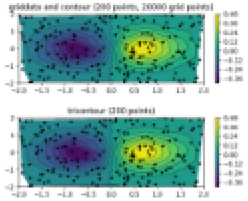


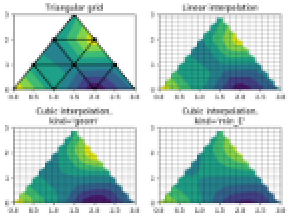
Figure Title



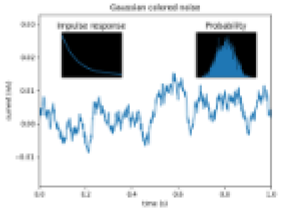
axhspan Demo



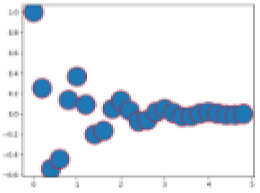
Tricontour Vs Griddata



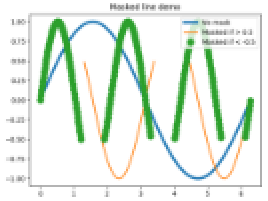
Triinterp Demo



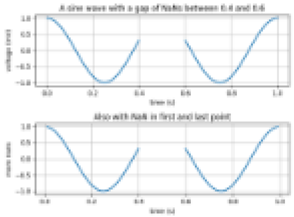
Axes Demo



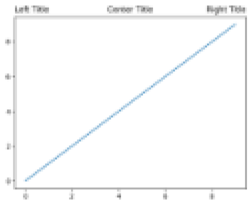
Arctest



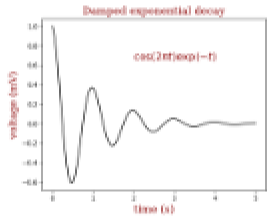
Masked Demo



Nan Test

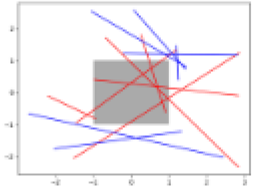


Titles Demo

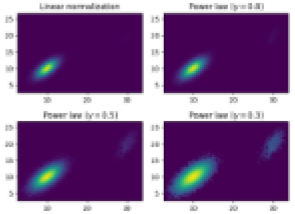


Controlling style of text and labels using a dictionary

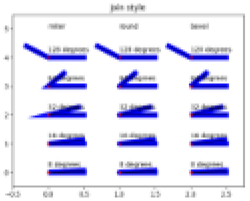
Other Plotting Samples



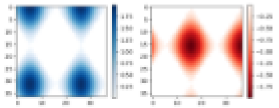
Changing colors of lines intersecting a box



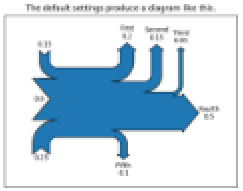
Exploring normalizations



Join styles

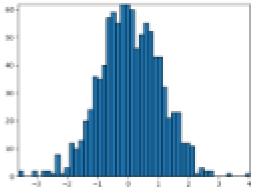


Colorbar

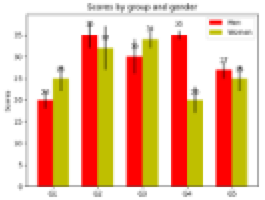


The Sankey class

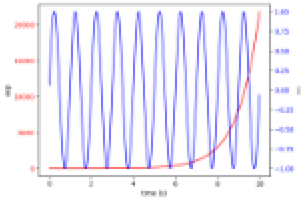
Matplotlib Logos



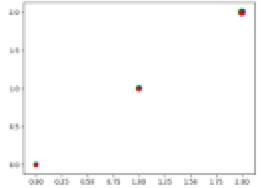
Building histograms using Rectangles and PolyCollections



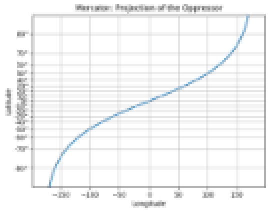
Barchart



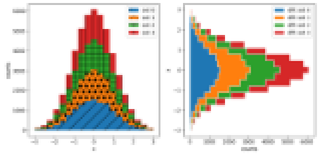
Plots with different scales



Scatter plot with pie chart markers



Custom scale



Hatch-filled histograms

Master in Software Engineering

Hussam Hourani has over 25 years of Organizations Transformation, VROs, PMO, Large Scale and Enterprise Programs Global Delivery, Leadership, Business Development and Management Consulting. His client experience is wide ranging across many sectors but focuses on Performance Enhancement, Transformation, Enterprise Program Management, Artificial Intelligence and Data Science.