In [ ]: # matplotlib programs (2-D)

In [2]: import matplotlib.pyplot as plt

import numpy as np

# Prepare the data

x = np.linspace(0, 10, 100)

# Plot the data

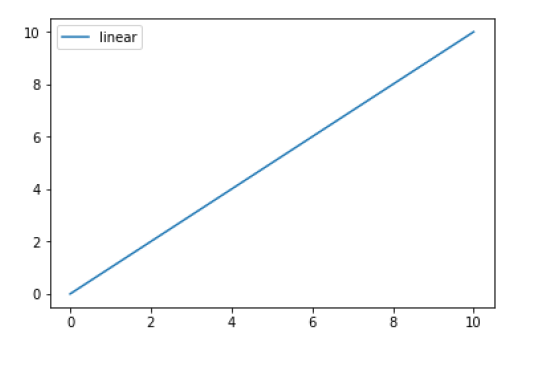
plt.plot(x, x, label='linear')

# Add a legend

plt.legend()

# Show the plot

plt.show()



In [6]: fig = plt.figure()

ax = fig.add\_subplot(111)

ax.plot([1, 2, 3, 4], [10, 20, 25, 30], color='lightblue', linew

idth=3)

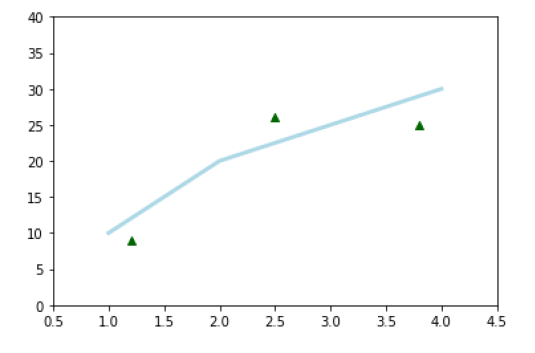
ax.scatter([0.3, 3.8, 1.2, 2.5], [11, 25, 9, 26], color='darkgre

en', marker='^')

ax.set\_xlim(0.5, 4.5)

ax.set\_ylim(0, 40)

plt.show()



In [7]: # Create a Figure

fig = plt.figure()

# Set up Axes

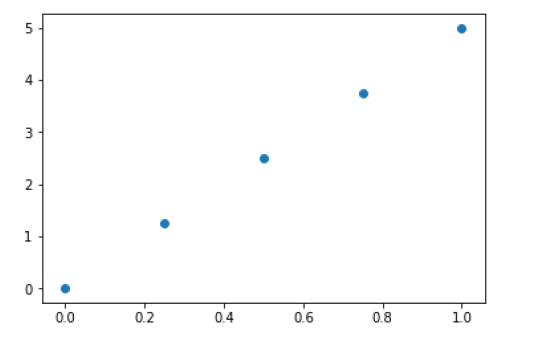
ax = fig.add\_subplot(111)

# Scatter the data

ax.scatter(np.linspace(0, 1, 5), np.linspace(0, 5, 5))

# Show the plot

plt.show()



In [7]: # Create a Figure

fig = plt.figure()

# Set up Axes

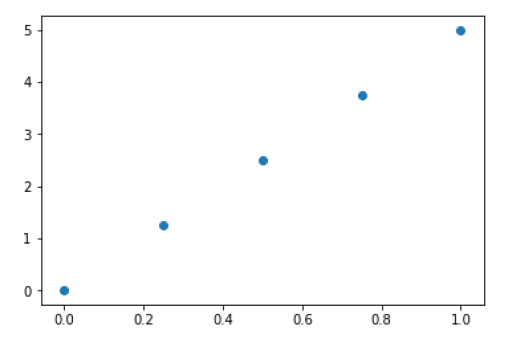
ax = fig.add\_subplot(111)

# Scatter the data

ax.scatter(np.linspace(0, 1, 5), np.linspace(0, 5, 5))

# Show the plot

plt.show()



In [8]: # Initialize the plot

fig = plt.figure(figsize=(20,10))

ax1 = fig.add\_subplot(121)

ax2 = fig.add\_subplot(122)

# or replace the three lines of code above by the following lin

e:

#fig, (ax1, ax2) = plt.subplots(1,2, figsize=(20,10))

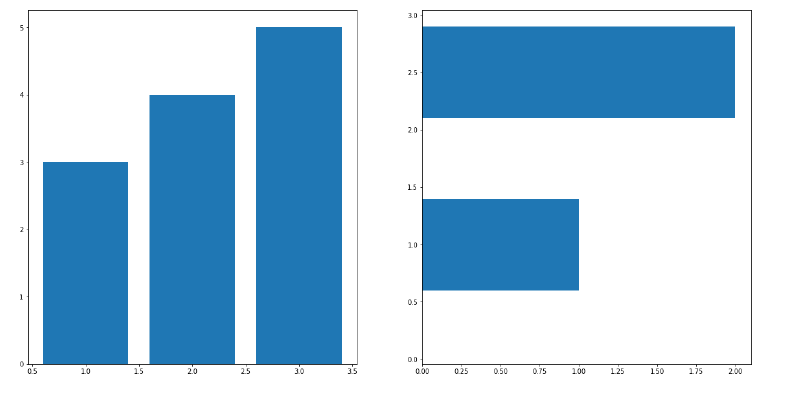
# Plot the data

ax1.bar([1,2,3],[3,4,5])

ax2.barh([0.5,1,2.5],[0,1,2])

# Show the plot

plt.show()



In [10]: # Initialize the plot

fig = plt.figure()

ax1 = fig.add\_subplot(131)

ax2 = fig.add\_subplot(132)

ax3 = fig.add\_subplot(133)

# Plot the data

ax1.bar([1,2,3],[3,4,5])

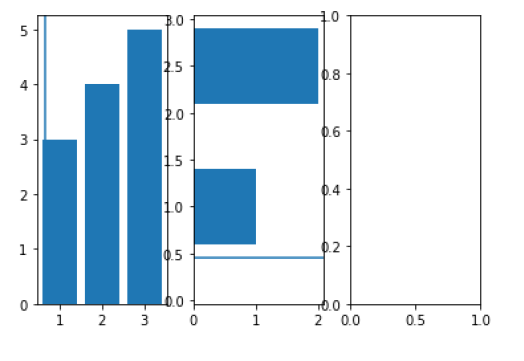
ax2.barh([0.5,1,2.5],[0,1,2])

ax2.axhline(0.45)

ax1.axvline(0.65)

# Show the plot

plt.show()



In [13]: ax = plt.subplot(111)

t = np.arange(0.0, 5.0, 0.01)

s = np.cos(2\*np.pi\*t)

line, = plt.plot(t, s, lw=2)

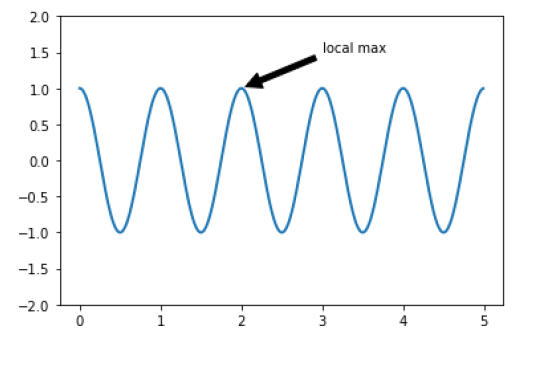
plt.annotate('local max', xy=(2, 1), xytext=(3, 1.5),

arrowprops=dict(facecolor='black', shrink=0.05),

)

plt.ylim(-2,2)

plt.show()



In [14]: from matplotlib.ticker import NullFormatter # useful for `logit` scale

# Fixing random state for reproducibility

np.random.seed(19680801)

# make up some data in the interval ]0, 1[

y = np.random.normal(loc=0.5, scale=0.4, size=1000)

y = y[(y > 0) & (y < 1)]

y.sort()

x = np.arange(len(y))

# plot with various axes scales

plt.figure(1)

# linear

plt.subplot(221)

plt.plot(x, y)

plt.yscale('linear')

plt.title('linear')

plt.grid(True)

# log

plt.subplot(222)

plt.plot(x, y)

plt.yscale('log')

plt.title('log')

plt.grid(True)

# symmetric log

plt.subplot(223)

plt.plot(x, y - y.mean())

plt.yscale('symlog', linthreshy=0.01)

plt.title('symlog')

plt.grid(True)

# logit

plt.subplot(224)

plt.plot(x, y)

plt.yscale('logit')

plt.title('logit')

plt.grid(True)

# Format the minor tick labels of the y-axis into empty strings

with

# `NullFormatter`, to avoid cumbering the axis with too many lab

els.

plt.gca().yaxis.set\_minor\_formatter(NullFormatter())

# Adjust the subplot layout, because the logit one may take more

space

# than usual, due to y-tick labels like "1 - 10^{-3}"

plt.subplots\_adjust(top=0.92, bottom=0.08, left=0.10, right=0.95

, hspace=0.25,

wspace=0.35)

plt.show()

