Matplotlib scatter plot legend

I created a 4D scatter plot graph to represent different temperatures in a specific area. When I create the legend, the legend shows the correct symbol and color but adds a line through it. The code I'm using is:

```
colors=['b', 'c', 'y', 'm', 'r']
lo = plt.Line2D(range(10), range(10), marker='x', color=colors[0])
ll = plt.Line2D(range(10), range(10), marker='o', color=colors[0])
l = plt.Line2D(range(10), range(10), marker='o',color=colors[1])
a = plt.Line2D(range(10), range(10), marker='o',color=colors[2])
h = plt.Line2D(range(10), range(10), marker='o',color=colors[3])
hh = plt.Line2D(range(10), range(10), marker='o',color=colors[4])
ho = plt.Line2D(range(10), range(10), marker='x', color=colors[4])
plt.legend((lo,ll,l,a, h, hh, ho),('Low Outlier', 'LoLo','Lo', 'Average', 'Hi', 'HiHi',
'High Outlier'),numpoints=1, loc='lower left', ncol=3, fontsize=8)
```

I tried changing `Line2D` to `Scatter` and `scatter`. `Scatter` returned an error and `scatter` changed the graph and returned an error.

With `scatter`, I changed the `range(10)` to the lists containing the data points. Each list contains either the x, y, or z variable.

```
lo = plt.scatter(xLOutlier, yLOutlier, zLOutlier, marker='x', color=colors[0])
ll = plt.scatter(xLoLo, yLoLo, zLoLo, marker='o', color=colors[0])
l = plt.scatter(xLo, yLo, zLo, marker='o',color=colors[1])
a = plt.scatter(xAverage, yAverage, zAverage, marker='o',color=colors[2])
h = plt.scatter(xHi, yHi, zHi, marker='o',color=colors[3])
hh = plt.scatter(xHiHi, yHiHi, zHiHi, marker='o',color=colors[4])
```

```
ho = plt.scatter(xHOutlier, yHOutlier, zHOutlier, marker='x', color=colors[4])
plt.legend((lo,ll,l,a, h, hh, ho),('Low Outlier', 'LoLo','Lo', 'Average', 'Hi', 'HiHi',
'High Outlier'),scatterpoints=1, loc='lower left', ncol=3, fontsize=8)
```

When I run this, the legend no longer exists, it is a small white box in the corner with nothing in it.

Any advice?

python matplotlib legend scatter-plot

edited Jan 21 at 14:09

hooy
4,435 1 16 33

asked Jul 1 '13 at 19:12

user2386081
127 2 3 8

I believe a much better solution is given here. – dmvianna Feb 6 '15 at 5:15

1 Answer

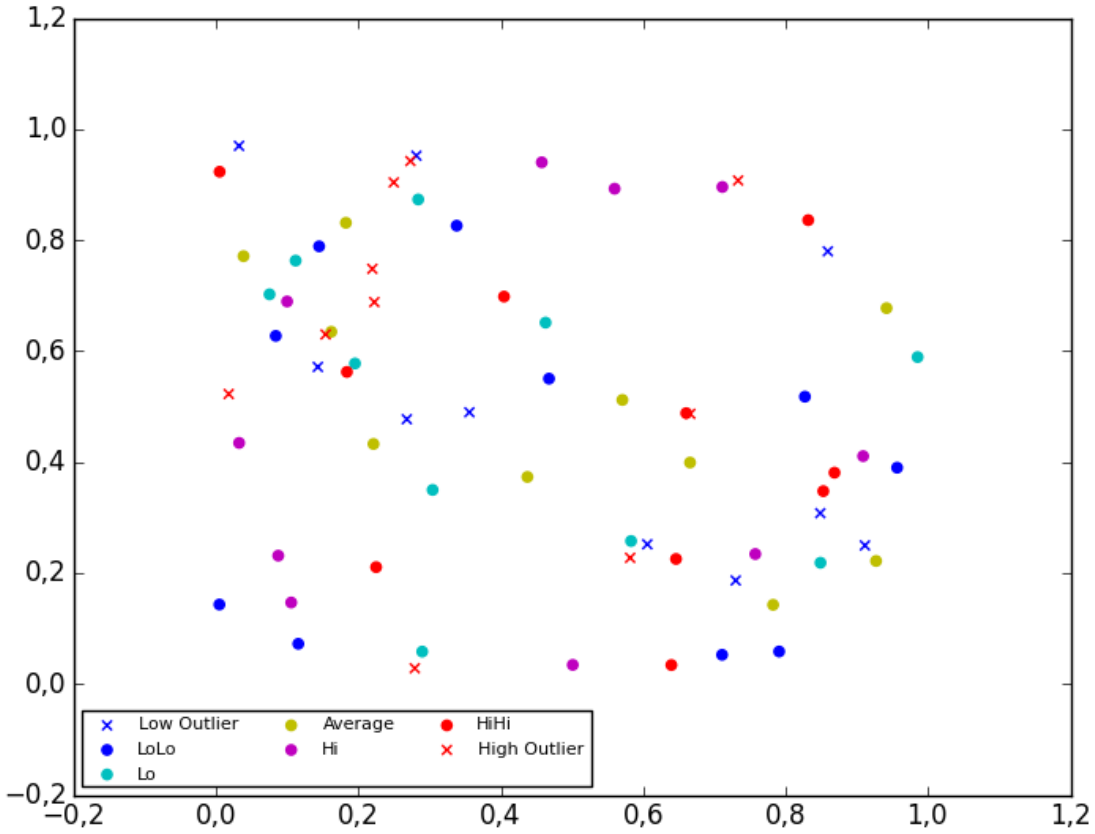
2D scatter plot

Using the `scatter` method of the `matplotlib.pyplot` module should work (at least with matplotlib 1.2.1 with Python 2.7.5), as in the example code below. Also, if you are using scatter plots, use `scatterpoints=1` rather than `numpoints=1` in the legend call to have only one point for each legend entry.

In the code below I've used random values rather than plotting the same range over and over, making all the plots visible (i.e. not overlapping each other).

```
import matplotlib.pyplot as plt
from numpy.random import random

colors = ['b', 'c', 'y', 'm', 'r']

lo = plt.scatter(random(10), random(10), marker='x', color=colors[0])
ll = plt.scatter(random(10), random(10), marker='o', color=colors[0])
l  = plt.scatter(random(10), random(10), marker='o', color=colors[1])
a  = plt.scatter(random(10), random(10), marker='o', color=colors[2])
h  = plt.scatter(random(10), random(10), marker='o', color=colors[3])
hh = plt.scatter(random(10), random(10), marker='o', color=colors[4])
```

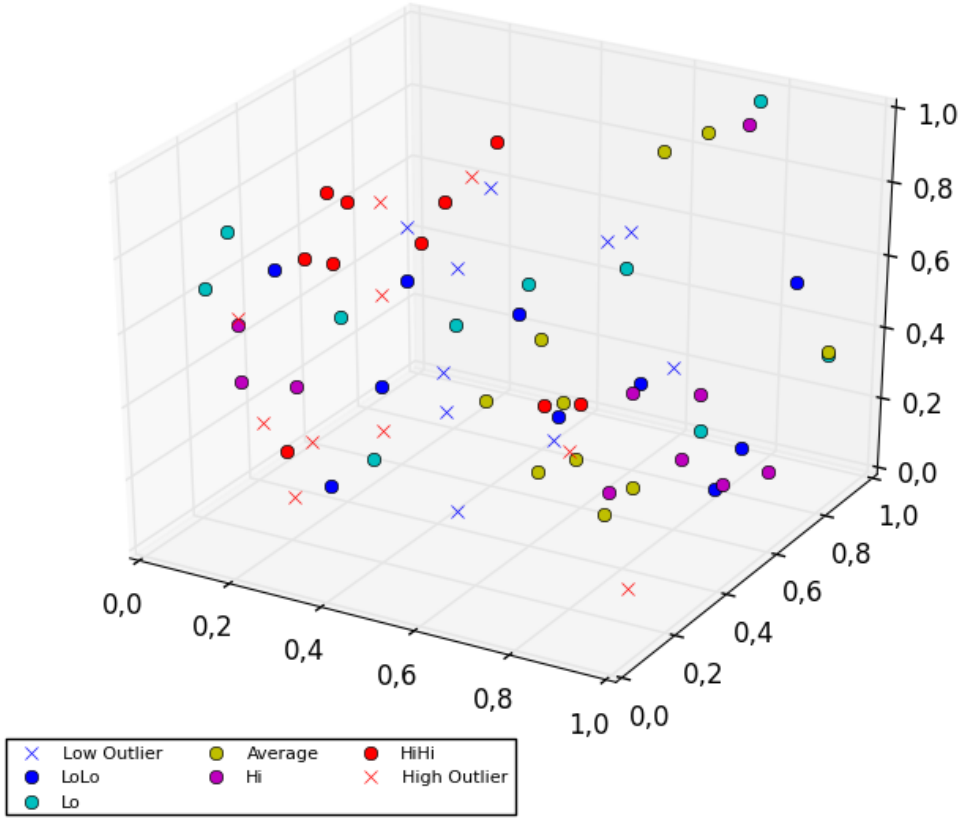
```
ho = plt.scatter(random(10), random(10), marker='x', color=colors[4])

plt.legend((lo, ll, l, a, h, hh, ho),
           ('Low Outlier', 'LoLo', 'Lo', 'Average', 'Hi', 'HiHi', 'High Outlier'),
           scatterpoints=1,
           loc='lower left',
           ncol=3,
           fontsize=8)

plt.show()
```



3D scatter plot

To plot a scatter in 3D, use the `plot` method, as the legend does not support `Patch3DCollection` as is returned by the `scatter` method of an `Axes3D` instance. To specify the markerstyle you can include this as a positional argument in the method call, as seen in the example below. Optionally one can include argument to both the `linestyle` and `marker` parameters.

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

colors=['b', 'c', 'y', 'm', 'r']

ax = plt.subplot(111, projection='3d')

ax.plot(random(10), random(10), random(10), 'x', color=colors[0], label='Low Outlier')
ax.plot(random(10), random(10), random(10), 'o', color=colors[0], label='LoLo')
ax.plot(random(10), random(10), random(10), 'o', color=colors[1], label='Lo')
ax.plot(random(10), random(10), random(10), 'o', color=colors[2], label='Average')
ax.plot(random(10), random(10), random(10), 'o', color=colors[3], label='Hi')
ax.plot(random(10), random(10), random(10), 'o', color=colors[4], label='HiHi')
ax.plot(random(10), random(10), random(10), 'x', color=colors[4], label='High Outlier')

plt.legend(loc='upper left', numpoints=1, ncol=3, fontsize=8, bbox_to_anchor=(0, 0))

plt.show()
```



edited Jul 2 '13 at 14:44

answered Jul 1 '13 at 19:35

hooy

4,435 1 16 33

Does this work for 3D scatter plots as well? – user2386081 Jul 1 '13 at 19:51

In order to get the legend to work when plotting a 3D scatter, use the `plot` method with the marker as a positional argument. See edit. – hooy Jul 2 '13 at 14:38