How to surface plot/3d plot from dataframe?

I am new to `pandas` and `matplotlib`. Couldn't able to get exact reference to plot my `DataFrame` whose schema is as follows

```
schema = StructType([
StructField("x", IntegerType(), True),
StructField("y", IntegerType(), True),
StructField("z", IntegerType(), True)])
```

Like to plot 3d graph w.r.t. x, y and z

Here is the sample code i used

```
import matplotlib.pyplot as pltt

dfSpark = sqlContext.createDataFrame(tupleRangeRDD, schema) // reading as spark df
df = dfSpark.toPandas()
fig = pltt.figure();
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(df['x'], df['y'], df['z'])
```

I am getting a empty graph plot. definitely missing something. Any pointers?

-Thx

Request-1: Print df

```
def print_full(x):
pd.set_option('display.max_rows', len(x))
print(x)
pd.reset_option('display.max_rows')


print_full(df)
```

Result of top 10

```
         x    y       z
0      301  301      10
1      300  301      16
2      300  300       6
3      299  301      30
4      299  300      20
5      299  299      14
6      298  301      40
7      298  300      30
8      298  299      24
9      298  298      10
10     297  301      48
```

python numpy pandas matplotlib dataframe

edited Jun 21 '16 at 14:49
Stefan
12.2k 4 17 39

asked Apr 13 '16 at 5:41
mohan
167 1 5 19

Does df contain anything? If so, can you print df.head(n=10) in your question? – giosans Apr 13 '16 at 7:08

update my question with printing df – mohan Apr 13 '16 at 14:43

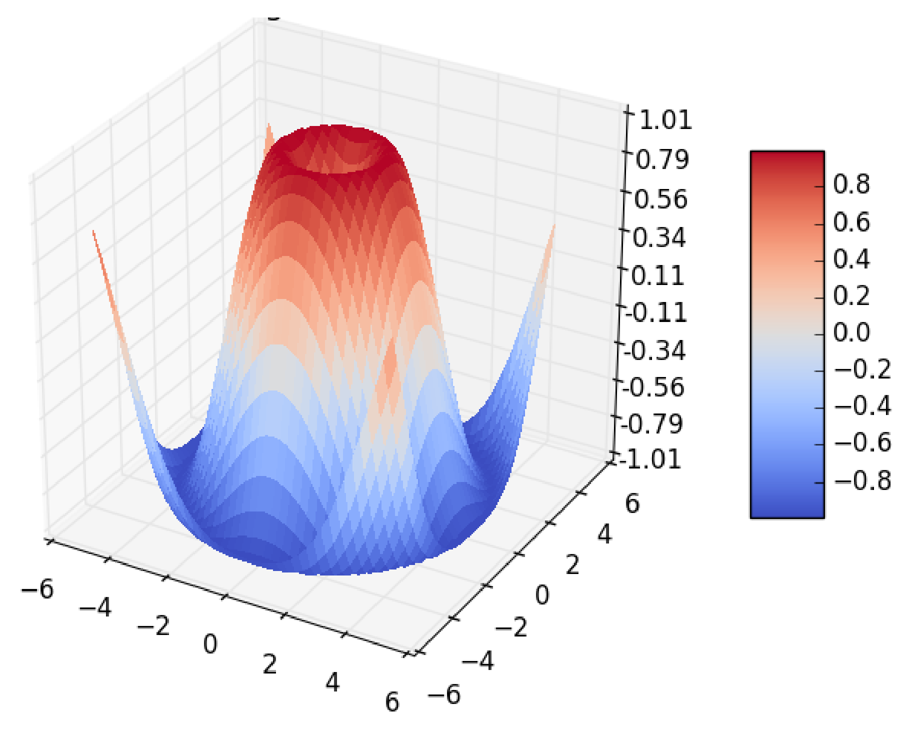
1 Answer

`.plot_surface()` takes `2D` `arrays` as inputs, not `1D` `DataFrame` columns. This has been explained quite well here, along with the below code that illustrates how one could arrive at the required format using `DataFrame` input. Reproduced below with minor modifications like additional comments.

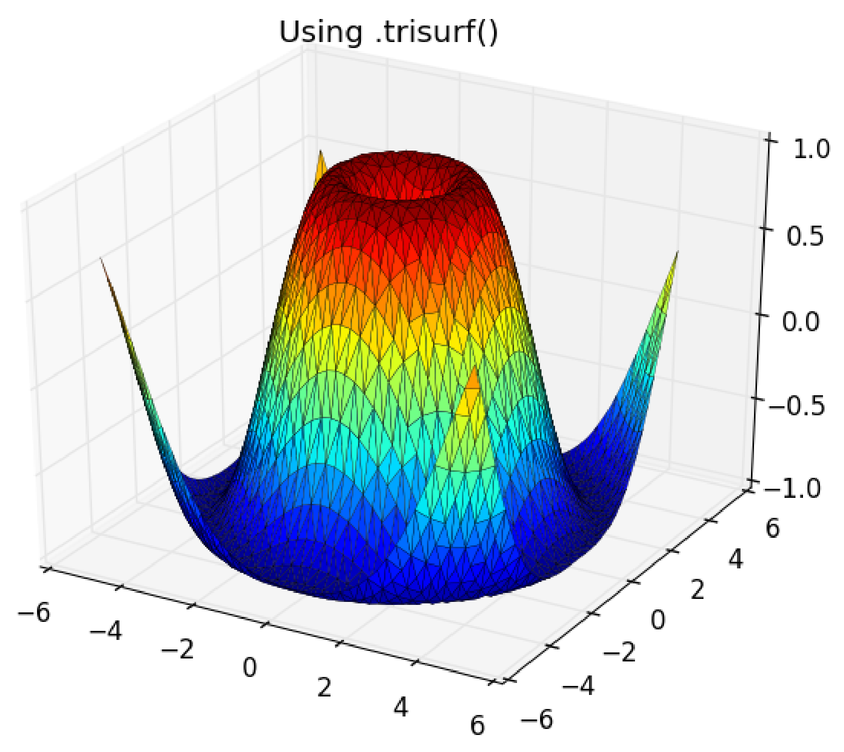
Alternatively, however, there is `.plot_trisurf()` which uses `1D` inputs. I've added an example in the middle of the code.

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm
from matplotlib.ticker import LinearLocator, FormatStrFormatter
from mpl_toolkits.mplot3d import Axes3D

## Matplotlib Sample Code using 2D arrays via meshgrid
X = np.arange(-5, 5, 0.25)
Y = np.arange(-5, 5, 0.25)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X ** 2 + Y ** 2)
Z = np.sin(R)
fig = plt.figure()
ax = Axes3D(fig)
surf = ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap=cm.coolwarm,
                       linewidth=0, antialiased=False)
ax.set_zlim(-1.01, 1.01)

ax.zaxis.set_major_locator(LinearLocator(10))
ax.zaxis.set_major_formatter(FormatStrFormatter('%.02f'))

fig.colorbar(surf, shrink=0.5, aspect=5)
plt.title('Original Code')
plt.show()
```

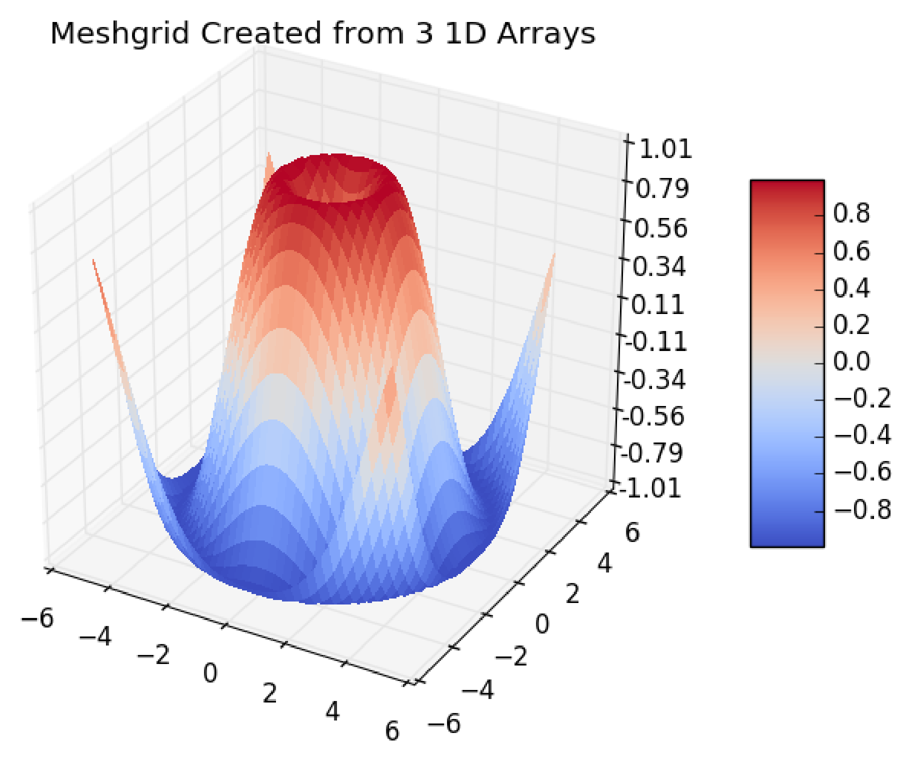
Original Code



```
## DataFrame from 2D-arrays
x = X.reshape(1600)
y = Y.reshape(1600)
z = Z.reshape(1600)
df = pd.DataFrame({'x': x, 'y': y, 'z': z}, index=range(len(x)))

# Plot using `.trisurf()`:

ax.plot_trisurf(df.x, df.y, df.z, cmap=cm.jet, linewidth=0.2)
plt.show()
```



```
# 2D-arrays from DataFrame
x1 = np.linspace(df['x'].min(), df['x'].max(), len(df['x'].unique()))
y1 = np.linspace(df['y'].min(), df['y'].max(), len(df['y'].unique()))

"""
x, y via meshgrid for vectorized evaluation of
2 scalar/vector fields over 2-D grids, given
one-dimensional coordinate arrays x1, x2,..., xn.
"""

x2, y2 = np.meshgrid(x1, y1)

# Interpolate unstructured D-dimensional data.
z2 = griddata((df['x'], df['y']), df['z'], (x2, y2), method='cubic')

# Ready to plot
fig = plt.figure()
ax = fig.gca(projection='3d')
surf = ax.plot_surface(x2, y2, z2, rstride=1, cstride=1, cmap=cm.coolwarm,
                       linewidth=0, antialiased=False)
ax.set_zlim(-1.01, 1.01)

ax.zaxis.set_major_locator(LinearLocator(10))
ax.zaxis.set_major_formatter(FormatStrFormatter('%.02f'))

fig.colorbar(surf, shrink=0.5, aspect=5)
plt.title('Meshgrid Created from 3 1D Arrays')

plt.show()
```



edited Feb 28 at 21:24
brent.payne
2,128 1 15 16

answered Apr 13 '16 at 15:00
Stefan
12.2k 4 17 39