In [ ]:

```
import pandas as pd
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt
from scipy.sparse.linalg import svds
from scipy import sparse
import seaborn as sns
from matplotlib.axis import Axis

#基于矩阵分解的结果，复原矩阵
def rebuildMatrix(U, sigma, V):
    a = np.dot(U, sigma)
    a = np.dot(a, np.transpose(V))
    return a

def GetMatrixImage(fig,fignum,vt,vt_name):
    for i in range(len(vt)-1):
        ax = fig.add_subplot(9,2,i*2+fignum)
        # Set tick font size
        for label in (ax.get_xticklabels() + ax.get_yticklabels()):
            label.set_fontsize(16)
        # Show ticks in the left and lower axes only
        Axis.set_label_coords(ax.yaxis,0, 0.5)
        Axis.set_label_coords(ax.xaxis,0.5, 0)
        ax.spines['bottom'].set_position(('data', 0))
        ax.spines['left'].set_position(('data', 0))
        g = sns.scatterplot( x="v{}".format(i+1), y="v{}".format(i+2),data=pd
            "v{}".format(i+1):vt[i],
            "v{}".format(i+2):vt[i+1]
        }),palette="Set2")
        img_title = "Spectral Subspace Plots of {0}{1} and {0}{2} ".format(vt_
        g.set_title(img_title,fontsize = 20)
        ax.set_xlabel(vt_name+str(i+1),fontsize = 20)
        ax.set_ylabel(vt_name+str(i+2),fontsize = 20)
        for p in ax.patches:
            height = p.get_height()
            ax.text(p.get_x()+p.get_width()/2.,
                    height + 3,
                    '{:1.2f}%'.format(100*height/len(vt)),
                    ha="center")
```

1.LSI

In [ ]:

```
data = [
    [1,0,1,0,0,0],
    [0,1,0,0,0,0],
    [1,1,0,0,0,0],
    [1,0,0,1,1,0],
    [0,0,0,1,0,1]]
data
```

Out[ ]:

```
[[1, 0, 1, 0, 0, 0],
 [0, 1, 0, 0, 0, 0],
 [1, 1, 0, 0, 0, 0],
 [1, 0, 0, 1, 1, 0],
 [0, 0, 0, 1, 0, 1]]
```

In [ ]:

```
yelp_matrix = sparse.coo_matrix(data, dtype=float)
print(yelp_matrix)
```

```
u, s, vt = svds(yelp_matrix,k=3,which = 'LM')
```

```
  (0, 0)	1.0
  (0, 2)	1.0
  (1, 1)	1.0
  (2, 0)	1.0
  (2, 1)	1.0
  (3, 0)	1.0
  (3, 3)	1.0
  (3, 4)	1.0
  (4, 3)	1.0
  (4, 5)	1.0
```

三维表示的特征值

In [ ]:

```
s_matric = np.diag(s)
print(s_matric)
```

```
[[1.27529025 0.         0.        ]
 [0.         1.59438237 0.        ]
 [0.         0.         2.16250096]]
```

词项的三维表示

In [ ]:

```
u
```

Out[ ]:

```
array([[-0.56949758,  0.29617436,  0.44034748],
       [ 0.5870217 ,  0.33145069,  0.12934635],
       [ 0.36768998,  0.51111524,  0.47553026],
       [-0.15490588, -0.35057241,  0.70302032],
       [ 0.4145917 , -0.64674677,  0.26267284]])
```

文档的三维表示

In [ ]:

```
vt.T
```

Out[ ]:

```
array([[-0.2797116 ,  0.28645399,  0.74862305],
       [ 0.74862305,  0.52845914,  0.2797116 ],
       [-0.44656311,  0.18576119,  0.2036288 ],
       [ 0.2036288 , -0.6255207 ,  0.44656311],
       [-0.12146715, -0.21987976,  0.32509596],
       [ 0.32509596, -0.40564094,  0.12146715]])
```

利用SVD表达的三维表示复原词项文档矩阵

In [ ]:

```
data_lsi = np.absolute(rebuildMatrix(u, s_matric, vt.T))
data_lsi
```

Out[ ]:

```
array([[1.05129307, 0.02780367, 0.60595265, 0.01803027, 0.29396119,
        0.31199146],
       [0.15137923, 0.91794411, 0.1791829 , 0.05321203, 0.11619747,
        0.06298543],
       [0.87211017, 1.06932334, 0.15137923, 0.04495516, 0.0981672 ,
        0.05321203],
       [1.0332628 , 0.01803027, 0.29396119, 0.98830764, 0.64113441,
        0.34717322],
       [0.01803027, 0.0097734 , 0.31199146, 1.00633791, 0.34717322,
        0.65916468]])
```