In [60]:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as py
from mpl_toolkits.mplot3d import Axes3D

# read data
dataset = pd.read_csv("diabetes.csv")

# header data
dataset.head()
```

Out[60]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunc
0	6	148	72	35	0	33.6	0.
1	1	85	66	29	0	26.6	0.
2	8	183	64	0	0	23.3	0.
3	1	89	66	23	94	28.1	0.
4	0	137	40	35	168	43.1	2.

In [61]:

```
# description of dataset features
dataset.describe()
```

Out[61]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	Diak
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	

In [62]:

```
X = dataset.iloc[: ,0:8]
print(X)
Y = dataset.iloc[: ,8]

# proportion of variance
X_std = (X - np.mean(X,axis=0))/np.std(X,axis=0)
print(X_std)
```

```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0              6      148             72             35        0  33.6
1              1       85             66             29        0  26.6
2              8      183             64              0        0  23.3
3              1       89             66             23       94  28.1
4              0      137             40             35      168  43.1
5              5      116             74              0        0  25.6
6              3       78             50             32       88  31.0
7             10      115              0              0        0  35.3
8              2      197             70             45      543  30.5
9              8      125             96              0        0   0.0
10             4      110             92              0        0  37.6
11            10      168             74              0        0  38.0
12            10      139             80              0        0  27.1
13             1      189             60             23      846  30.1
14             5      166             72             19      175  25.8
15             7      100              0              0        0  30.0
16             0      118             84             47      230  45.8
17             7      107             74              0        0  29.6
18             1      103             30             38       83  43.3
19             1      115             70             30       96  34.6
20             3      126             88             41      235  39.3
21             8       99             84              0        0  35.4
22             7      196             90              0        0  39.8
23             9      119             80             35        0  29.0
24            11      143             94             33      146  36.6
25            10      125             70             26      115  31.1
26             7      147             76              0        0  39.4
27             1       97             66             15      140  23.2
28            13      145             82             19      110  22.2
29             5      117             92              0        0  34.1
..           ...      ...            ...            ...      ...   ...
738            2       99             60             17      160  36.6
739            1      102             74              0        0  39.5
740           11      120             80             37      150  42.3
741            3      102             44             20       94  30.8
742            1      109             58             18      116  28.5
743            9      140             94              0        0  32.7
744           13      153             88             37      140  40.6
745           12      100             84             33      105  30.0
746            1      147             94             41        0  49.3
747            1       81             74             41       57  46.3
748            3      187             70             22      200  36.4
749            6      162             62              0        0  24.3
750            4      136             70              0        0  31.2
751            1      121             78             39       74  39.0
752            3      108             62             24        0  26.0
753            0      181             88             44      510  43.3
754            8      154             78             32        0  32.4
755            1      128             88             39      110  36.5
756            7      137             90             41        0  32.0
757            0      123             72              0        0  36.3
758            1      106             76              0        0  37.5
759            6      190             92              0        0  35.5
760            2       88             58             26       16  28.4
761            9      170             74             31        0  44.0
762            9       89             62              0        0  22.5
763           10      101             76             48      180  32.9
764            2      122             70             27        0  36.8
765            5      121             72             23      112  26.2
766            1      126             60              0        0  30.1
```

```
767            1       93           70             31         0  30.4

     DiabetesPedigreeFunction  Age
0                       0.627   50
1                       0.351   31
2                       0.672   32
3                       0.167   21
4                       2.288   33
5                       0.201   30
6                       0.248   26
7                       0.134   29
8                       0.158   53
9                       0.232   54
10                      0.191   30
11                      0.537   34
12                      1.441   57
13                      0.398   59
14                      0.587   51
15                      0.484   32
16                      0.551   31
17                      0.254   31
18                      0.183   33
19                      0.529   32
20                      0.704   27
21                      0.388   50
22                      0.451   41
23                      0.263   29
24                      0.254   51
25                      0.205   41
26                      0.257   43
27                      0.487   22
28                      0.245   57
29                      0.337   38
..                        ...  ...
738                     0.453   21
739                     0.293   42
740                     0.785   48
741                     0.400   26
742                     0.219   22
743                     0.734   45
744                     1.174   39
745                     0.488   46
746                     0.358   27
747                     1.096   32
748                     0.408   36
749                     0.178   50
750                     1.182   22
751                     0.261   28
752                     0.223   25
753                     0.222   26
754                     0.443   45
755                     1.057   37
756                     0.391   39
757                     0.258   52
758                     0.197   26
759                     0.278   66
760                     0.766   22
761                     0.403   43
762                     0.142   33
763                     0.171   63
764                     0.340   27
```

```
765                    0.245   30
766                    0.349   47
767                    0.315   23

[768 rows x 8 columns]
     Pregnancies   Glucose  BloodPressure  SkinThickness   Insulin        B
MI  \
0       0.639947  0.848324       0.149641       0.907270 -0.692891  0.2040
13
1      -0.844885 -1.123396      -0.160546       0.530902 -0.692891 -0.6844
22
2       1.233880  1.943724      -0.263941      -1.288212 -0.692891 -1.1032
55
3      -0.844885 -0.998208      -0.160546       0.154533  0.123302 -0.4940
43
4      -1.141852  0.504055      -1.504687       0.907270  0.765836  1.4097
46
5       0.342981 -0.153185       0.253036      -1.288212 -0.692891 -0.8113
41
6      -0.250952 -1.342476      -0.987710       0.719086  0.071204 -0.1259
77
7       1.827813 -0.184482      -3.572597      -1.288212 -0.692891  0.4197
75
8      -0.547919  2.381884       0.046245       1.534551  4.021922 -0.1894
37
9       1.233880  0.128489       1.390387      -1.288212 -0.692891 -4.0604
74
10      0.046014 -0.340968       1.183596      -1.288212 -0.692891  0.7116
90
11      1.827813  1.474267       0.253036      -1.288212 -0.692891  0.7624
57
12      1.827813  0.566649       0.563223      -1.288212 -0.692891 -0.6209
62
13     -0.844885  2.131507      -0.470732       0.154533  6.652839 -0.2402
05
14      0.342981  1.411672       0.149641      -0.096379  0.826616 -0.7859
57
15      0.936914 -0.653939      -3.572597      -1.288212 -0.692891 -0.2528
97
16     -1.141852 -0.090591       0.770014       1.660007  1.304175  1.7524
28
17      0.936914 -0.434859       0.253036      -1.288212 -0.692891 -0.3036
64
18     -0.844885 -0.560048      -2.021665       1.095454  0.027790  1.4351
29
19     -0.844885 -0.184482       0.046245       0.593630  0.140667  0.3309
32
20     -0.250952  0.159787       0.976805       1.283638  1.347590  0.9274
52
21      1.233880 -0.685236       0.770014      -1.288212 -0.692891  0.4324
67
22      0.936914  2.350587       1.080200      -1.288212 -0.692891  0.9909
12
23      1.530847 -0.059293       0.563223       0.907270 -0.692891 -0.3798
16
24      2.124780  0.691838       1.286991       0.781814  0.574812  0.5847
71
25      1.827813  0.128489       0.046245       0.342717  0.305642 -0.1132
85
26      0.936914  0.817027       0.356432      -1.288212 -0.692891  0.9401
44
```

```
27      -0.844885 -0.747831      -0.160546      -0.347291  0.522715 -1.1159
47
28       2.718712  0.754432       0.666618      -0.096379  0.262228 -1.2428
67
29       0.342981 -0.121888       1.183596      -1.288212 -0.692891  0.2674
72
..            ...       ...            ...            ...       ...
...
738     -0.547919 -0.685236      -0.470732      -0.221835  0.696373  0.5847
71
739     -0.844885 -0.591345       0.253036      -1.288212 -0.692891  0.9528
36
740      2.124780 -0.027996       0.563223       1.032726  0.609544  1.3082
10
741     -0.250952 -0.591345      -1.297896      -0.033651  0.123302 -0.1513
61
742     -0.844885 -0.372265      -0.574128      -0.159107  0.314325 -0.4432
75
743      1.530847  0.597947       1.286991      -1.288212 -0.692891  0.0897
85
744      2.718712  1.004810       0.976805       1.032726  0.522715  1.0924
47
745      2.421746 -0.653939       0.770014       0.781814  0.218813 -0.2528
97
746     -0.844885  0.817027       1.286991       1.283638 -0.692891  2.1966
45
747     -0.844885 -1.248585       0.253036       1.283638 -0.197966  1.8158
87
748     -0.250952  2.068912       0.046245       0.091805  1.043689  0.5593
87
749      0.639947  1.286484      -0.367337      -1.288212 -0.692891 -0.9763
36
750      0.046014  0.472758       0.046245      -1.288212 -0.692891 -0.1005
93
751     -0.844885  0.003301       0.459827       1.158182 -0.050356  0.8893
77
752     -0.250952 -0.403562      -0.367337       0.217261 -0.692891 -0.7605
73
753     -1.141852  1.881130       0.976805       1.471822  3.735386  1.4351
29
754      1.233880  1.036107       0.459827       0.719086 -0.692891  0.0517
10
755     -0.844885  0.222381       0.976805       1.158182  0.262228  0.5720
79
756      0.936914  0.504055       1.080200       1.283638 -0.692891  0.0009
42
757     -1.141852  0.065895       0.149641      -1.288212 -0.692891  0.5466
95
758     -0.844885 -0.466156       0.356432      -1.288212 -0.692891  0.6989
98
759      0.639947  2.162804       1.183596      -1.288212 -0.692891  0.4451
59
760     -0.547919 -1.029505      -0.574128       0.342717 -0.553964 -0.4559
67
761      1.530847  1.536861       0.253036       0.656358 -0.692891  1.5239
73
762      1.530847 -0.998208      -0.367337      -1.288212 -0.692891 -1.2047
91
763      1.827813 -0.622642       0.356432       1.722735  0.870031  0.1151
69
764     -0.547919  0.034598       0.046245       0.405445 -0.692891  0.6101
```

```
54
765     0.342981  0.003301       0.149641       0.154533  0.279594 -0.7351
90
766    -0.844885  0.159787      -0.470732      -1.288212 -0.692891 -0.2402
05
767    -0.844885 -0.873019       0.046245       0.656358 -0.692891 -0.2021
29

     DiabetesPedigreeFunction       Age
0                    0.468492  1.425995
1                   -0.365061 -0.190672
2                    0.604397 -0.105584
3                   -0.920763 -1.041549
4                    5.484909 -0.020496
5                   -0.818079 -0.275760
6                   -0.676133 -0.616111
7                   -1.020427 -0.360847
8                   -0.947944  1.681259
9                   -0.724455  1.766346
10                  -0.848280 -0.275760
11                   0.196681  0.064591
12                   2.926869  2.021610
13                  -0.223115  2.191785
14                   0.347687  1.511083
15                   0.036615 -0.105584
16                   0.238963 -0.190672
17                  -0.658012 -0.190672
18                  -0.872441 -0.020496
19                   0.172520 -0.105584
20                   0.701041 -0.531023
21                  -0.253316  1.425995
22                  -0.063049  0.660206
23                  -0.630831 -0.360847
24                  -0.658012  1.511083
25                  -0.805998  0.660206
26                  -0.648952  0.830381
27                   0.045675 -0.956462
28                  -0.685193  2.021610
29                  -0.407342  0.404942
..                        ...       ...
738                 -0.057009 -1.041549
739                 -0.540228  0.745293
740                  0.945671  1.255820
741                 -0.217075 -0.616111
742                 -0.763716 -0.956462
743                  0.791645  1.000557
744                  2.120497  0.490030
745                  0.048695  1.085644
746                 -0.343920 -0.531023
747                  1.884928 -0.105584
748                 -0.192914  0.234767
749                 -0.887541  1.425995
750                  2.144658 -0.956462
751                 -0.636871 -0.445935
752                 -0.751636 -0.701198
753                 -0.754656 -0.616111
754                 -0.087210  1.000557
755                  1.767143  0.319855
756                 -0.244256  0.490030
757                 -0.645932  1.596171
758                 -0.830159 -0.616111
```

```
759                 -0.585529  2.787399
760                  0.888288 -0.956462
761                 -0.208015  0.830381
762                 -0.996266 -0.020496
763                 -0.908682  2.532136
764                 -0.398282 -0.531023
765                 -0.685193 -0.275760
766                 -0.371101  1.170732
767                 -0.473785 -0.871374

[768 rows x 8 columns]
```

In [79]:

```
covr_matrix = np.cov(X_std , rowvar = False)

covr_matrix1 = np.cov(X_std , rowvar = True)
print("covr_matrix.shape > "+str(covr_matrix.shape))
print("covr_matrix.shape1 > "+str(covr_matrix1.shape))

print("covariance of the matrix is : \n%s"  %covr_matrix1)
```

```
covr_matrix.shape > (8, 8)
covr_matrix.shape1 > (768, 768)
covariance of the matrix is :
[[ 0.39975557  0.08646692  0.20797775 ... -0.0421051   0.22218062
  -0.0138336 ]
 [ 0.08646692  0.26330973 -0.38401385 ...  0.00550964 -0.09355759
   0.22232633]
 [ 0.20797775 -0.38401385  1.29856214 ...  0.06497533  0.22210676
  -0.42212296]
 ...
 [-0.0421051   0.00550964  0.06497533 ...  0.17925243 -0.12732675
   0.00339104]
 [ 0.22218062 -0.09355759  0.22210676 ... -0.12732675  0.5480178
  -0.23678779]
 [-0.0138336   0.22232633 -0.42212296 ...  0.00339104 -0.23678779
   0.29841243]]
```

In [85]:

```
eigenvalues , eigenvectors = np.linalg.eig(covr_matrix)
print("np.linalg.eig(covr_matrix) > " + str(len(np.linalg.eig(covr_matrix))))
print("the eigenvalues of the cov(x) matrix are : \n %s" %eigenvalues)
print("\n the eigenvectors of the cov(x) matrix are : \n%s" %eigenvectors)
```

```
np.linalg.eig(covr_matrix > 2
the eigenvalues of the cov(x) matrix are :
[2.09711056 1.73346726 0.42036353 0.40498938 0.68351839 0.76333832
 0.87667054 1.03097228]

 the eigenvectors of the cov(x) matrix are :
[[-0.1284321  -0.59378583 -0.58879003  0.11784098 -0.19359817  0.47560573
  -0.08069115  0.01308692]
 [-0.39308257 -0.17402908 -0.06015291  0.45035526 -0.09416176 -0.46632804
   0.40432871 -0.46792282]
 [-0.36000261 -0.18389207 -0.19211793 -0.01129554  0.6341159  -0.32795306
  -0.05598649  0.53549442]
 [-0.43982428  0.33196534  0.28221253  0.5662838  -0.00958944  0.48786206
  -0.03797608  0.2376738 ]
 [-0.43502617  0.25078106 -0.13200992 -0.54862138  0.27065061  0.34693481
   0.34994376 -0.33670893]
 [-0.45194134  0.1009598  -0.03536644 -0.34151764 -0.68537218 -0.25320376
  -0.05364595  0.36186463]
 [-0.27061144  0.122069   -0.08609107 -0.00825873  0.08578409 -0.11981049
  -0.8336801  -0.43318905]
 [-0.19802707 -0.62058853  0.71208542 -0.21166198  0.03335717  0.10928996
  -0.0712006  -0.07524755]]
```

In [92]:

```
eig_pairs = [(eigenvalues[index] , eigenvectors[: , index]) for index in range(len(eige
nvalues))]

print("-------\n")
print(eig_pairs)
print("-------\n")

eig_pairs.sort()
eig_pairs.reverse()
print("\n%s" %eig_pairs)
eig_value_sorted = [eig_pairs[index][0] for index in range(len(eigenvalues))]
eig_vector_sorted = [eig_pairs[index][1] for index in range(len(eigenvalues))]
print("Eigen values in sorted : \n%s" %eig_value_sorted)
print("Respective eigen vectors are : \n%s" %eig_vector_sorted)

print("*******\n")
print(eig_pairs.size())
print("*******\n")
```

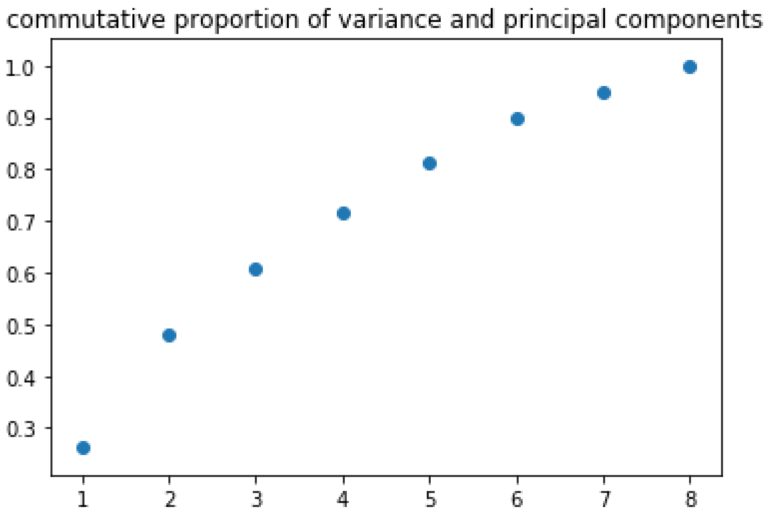
```
-------

[(2.0971105579945255, array([-0.1284321 , -0.39308257, -0.36000261, -0.439
82428, -0.43502617,
       -0.45194134, -0.27061144, -0.19802707])), (1.7334672594471274, arra
y([-0.59378583, -0.17402908, -0.18389207,  0.33196534,  0.25078106,
        0.1009598 ,  0.122069  , -0.62058853])), (0.420363528049568, array
([-0.58879003, -0.06015291, -0.19211793,  0.28221253, -0.13200992,
       -0.03536644, -0.08609107,  0.71208542])), (0.40498937781489885, arr
ay([ 0.11784098,  0.45035526, -0.01129554,  0.5662838 , -0.54862138,
       -0.34151764, -0.00825873, -0.21166198])), (0.6835183858447286, arra
y([-0.19359817, -0.09416176,  0.6341159 , -0.00958944,  0.27065061,
       -0.68537218,  0.08578409,  0.03335717])), (0.763338315649671, array
([ 0.47560573, -0.46632804, -0.32795306,  0.48786206,  0.34693481,
       -0.25320376, -0.11981049,  0.10928996])), (0.8766705419094799, arra
y([-0.08069115,  0.40432871, -0.05598649, -0.03797608,  0.34994376,
       -0.05364595, -0.8336801 , -0.0712006 ])), (1.0309722810083821, arra
y([ 0.01308692, -0.46792282,  0.53549442,  0.2376738 , -0.33670893,
        0.36186463, -0.43318905, -0.07524755]))]
-------



[(2.0971105579945255, array([-0.1284321 , -0.39308257, -0.36000261, -0.439
82428, -0.43502617,
       -0.45194134, -0.27061144, -0.19802707])), (1.7334672594471274, arra
y([-0.59378583, -0.17402908, -0.18389207,  0.33196534,  0.25078106,
        0.1009598 ,  0.122069  , -0.62058853])), (1.0309722810083821, arra
y([ 0.01308692, -0.46792282,  0.53549442,  0.2376738 , -0.33670893,
        0.36186463, -0.43318905, -0.07524755])), (0.8766705419094799, arra
y([-0.08069115,  0.40432871, -0.05598649, -0.03797608,  0.34994376,
       -0.05364595, -0.8336801 , -0.0712006 ])), (0.763338315649671, array
([ 0.47560573, -0.46632804, -0.32795306,  0.48786206,  0.34693481,
       -0.25320376, -0.11981049,  0.10928996])), (0.6835183858447286, arra
y([-0.19359817, -0.09416176,  0.6341159 , -0.00958944,  0.27065061,
       -0.68537218,  0.08578409,  0.03335717])), (0.420363528049568, array
([-0.58879003, -0.06015291, -0.19211793,  0.28221253, -0.13200992,
       -0.03536644, -0.08609107,  0.71208542])), (0.40498937781489885, arr
ay([ 0.11784098,  0.45035526, -0.01129554,  0.5662838 , -0.54862138,
       -0.34151764, -0.00825873, -0.21166198]))]
Eigen values in sorted :
[2.0971105579945255, 1.7334672594471274, 1.0309722810083821, 0.87667054190
94799, 0.763338315649671, 0.6835183858447286, 0.420363528049568, 0.4049893
7781489885]
Respective eigen vectors are :
[array([-0.1284321 , -0.39308257, -0.36000261, -0.43982428, -0.43502617,
       -0.45194134, -0.27061144, -0.19802707]), array([-0.59378583, -0.174
02908, -0.18389207,  0.33196534,  0.25078106,
        0.1009598 ,  0.122069  , -0.62058853]), array([ 0.01308692, -0.467
92282,  0.53549442,  0.2376738 , -0.33670893,
        0.36186463, -0.43318905, -0.07524755]), array([-0.08069115,  0.404
32871, -0.05598649, -0.03797608,  0.34994376,
       -0.05364595, -0.8336801 , -0.0712006 ]), array([ 0.47560573, -0.466
32804, -0.32795306,  0.48786206,  0.34693481,
       -0.25320376, -0.11981049,  0.10928996]), array([-0.19359817, -0.094
16176,  0.6341159 , -0.00958944,  0.27065061,
       -0.68537218,  0.08578409,  0.03335717]), array([-0.58879003, -0.060
15291, -0.19211793,  0.28221253, -0.13200992,
       -0.03536644, -0.08609107,  0.71208542]), array([ 0.11784098,  0.450
35526, -0.01129554,  0.5662838 , -0.54862138,
       -0.34151764, -0.00825873, -0.21166198])]
```

```
*******


---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-92-3309b4bf7200> in <module>
     14 
     15 print("*******\n")
---> 16 print(eig_pairs.size())
     17 print("*******\n")

AttributeError: 'list' object has no attribute 'size'
```

In [67]:

```
prop_vari = np.cumsum(eig_value_sorted)/sum(eig_value_sorted)
print("commutative proportion of variance : \n%s" %prop_vari)
num_comp = range(1 , len(eig_value_sorted)+1)
py.title("commutative proportion of variance and principal components")
#py.xlable("principal components")
#py.ylable("commutative proportion of variance")
py.scatter(num_comp , prop_vari)
py.show()
```

```
commutative proportion of variance :
[0.26179749 0.47819876 0.60690249 0.71634362 0.81163667 0.89696522
 0.94944224 1.        ]
```



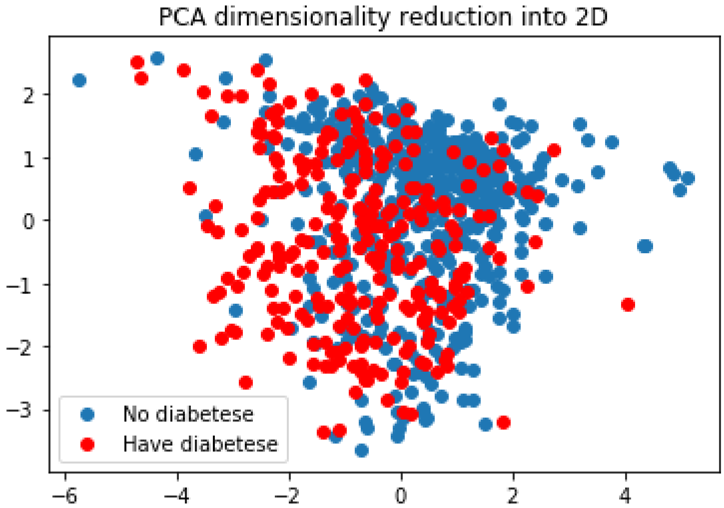
In [68]:

```
PCA_reduced = np.array(eig_vector_sorted[0:2]).transpose()
proj_data = np.dot(X_std , PCA_reduced)
print(proj_data)
```

```
[[-1.06850273 -1.23489499]
 [ 1.12168331  0.73385167]
 [ 0.39647671 -1.59587594]
 ...
 [ 0.28347525 -0.09706503]
 [ 1.06032431 -0.83706234]
 [ 0.83989172  1.15175485]]
```

In [69]:

```
negative = py.scatter(proj_data[: , 0][Y==0] , proj_data[: ,1][Y==0])
positive = py.scatter(proj_data[: , 0][Y==1] , proj_data[: ,1][Y==1] , color ='red')
py.title("PCA dimensionality reduction into 2D")
py.legend([negative , positive] , ["No diabetese" , "Have diabetese"])
py.show()
```



In [70]:

```
PCA_reduced = np.array(eig_vector_sorted[0:3]).transpose()
proj_data_3D = np.dot(X_std , PCA_reduced)
print(proj_data_3D)
```

```
[[-1.06850273 -1.23489499 -0.09592984]
 [ 1.12168331  0.73385167  0.71293816]
 [ 0.39647671 -1.59587594 -1.76067844]
 ...
 [ 0.28347525 -0.09706503  0.07719194]
 [ 1.06032431 -0.83706234 -0.42503045]
 [ 0.83989172  1.15175485  1.00917817]]
```

In [71]:

```
fig = py.figure()
ax = Axes3D(fig)
negative = py.scatter(proj_data_3D[: , 0][Y==0] ,proj_data_3D[: , 1][Y==0] , proj_data_
3D[: , 2][Y==0])
positive = py.scatter(proj_data_3D[: , 0][Y==1] , proj_data_3D[: , 1][Y==1] , proj_data
_3D[: , 2][Y==1] , color ="red")
ax.set_title("PCA reduced data to 3D")
py.legend([negative , positive] , ["No diabetese" , "Have diabetese"])
py.show()
```