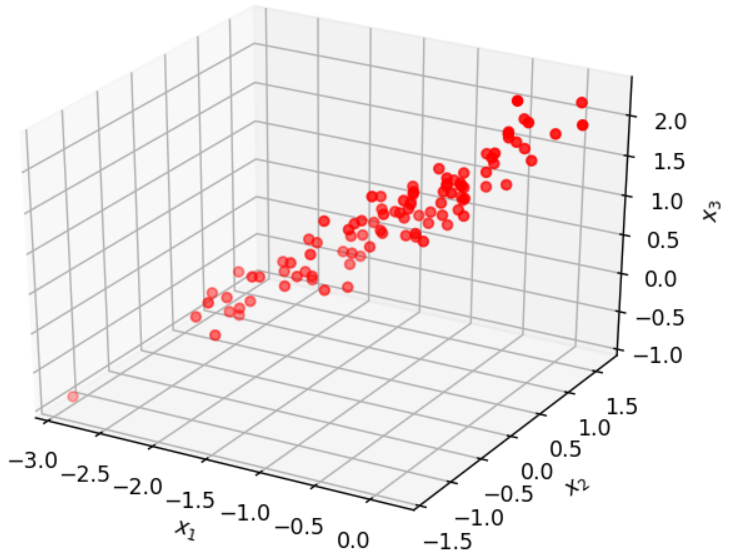
In [1]:
```
# Enable interactive rotation of graph
%matplotlib notebook

import numpy as np
from scipy.io import loadmat
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Load data for activity
X = loadmat('PCA_Activity.mat')['X']
rows, cols = np.array(X.shape)
x, y, z = X

print('Rows of X = ',rows)
print('Cols of X = ',cols)
```

```
Rows of X =  3
Cols of X =  100
```

In [2]:
```
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.scatter(x, y, z, c='r', marker='o')

ax.set_xlabel('$x_1$')
ax.set_ylabel('$x_2$')
ax.set_zlabel('$x_3$')

plt.show()
```

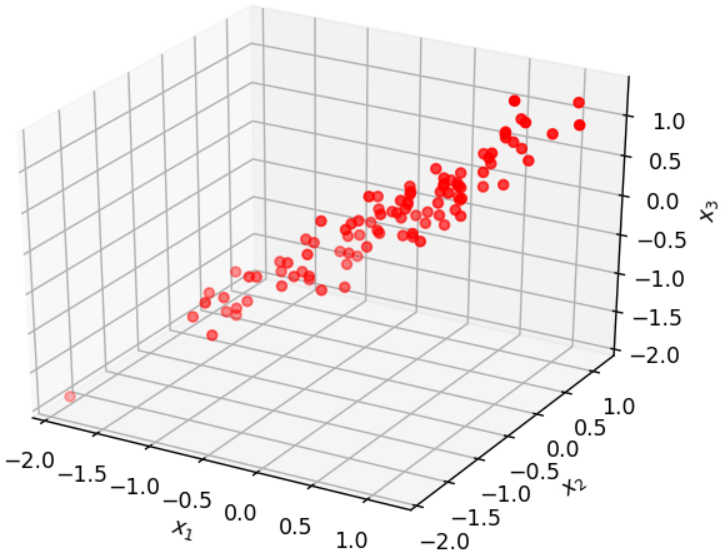
Figure 1



In [3]:
```
# Subtract mean
X_m = X - np.mean(X, 1).reshape((3,1))
x_m, y_m, z_m = X_m
```

In [4]:
```
# display zero mean scatter plot

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.scatter(x_m, y_m, z_m, c='r', marker='o')

ax.set_xlabel('$x_1$')
ax.set_ylabel('$x_2$')
ax.set_zlabel('$x_3$')

plt.show()
```

Figure 2

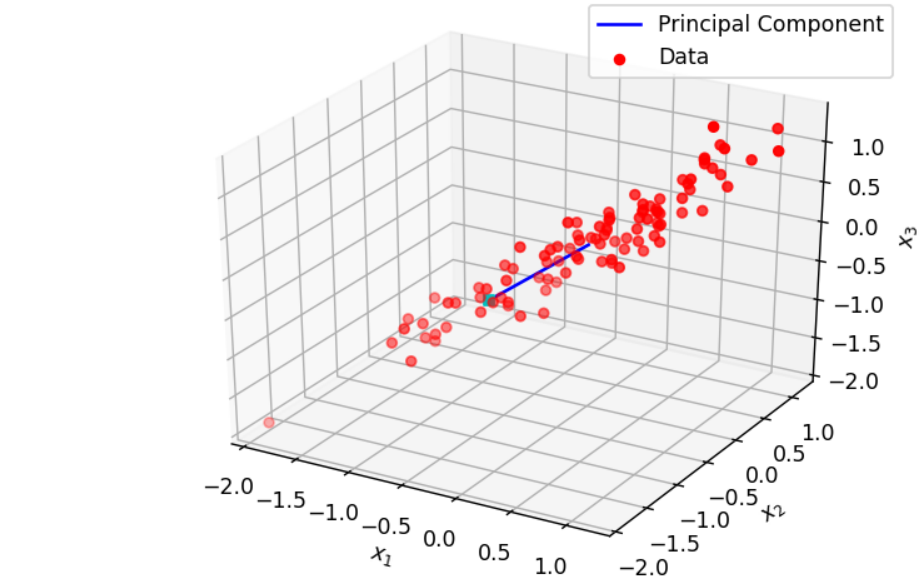


In [5]:
```
# Use SVD to find first principal component

U,s,VT = np.linalg.svd(X_m,full_matrices=False)

# complete the next line of code to assign the first principal component to a
a = U[:,[0]]

print(a)
```

```
[[-0.58277194]
 [-0.57701087]
 [-0.57221964]]
```

In [6]:
```
# display zero mean scatter plot and first principal component

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.scatter(x_m, y_m, z_m, c='r', marker='o', label='Data')

ax.scatter(a[0],a[1],a[2], c='c', marker='s')

ax.set_xlabel('$x_1$')
ax.set_ylabel('$x_2$')
ax.set_zlabel('$x_3$')

ax.plot([0,a[0]],[0,a[1]],[0,a[2]], c='b',label='Principal Component')

ax.legend()
plt.show()
```

Figure 3



Forward to next view

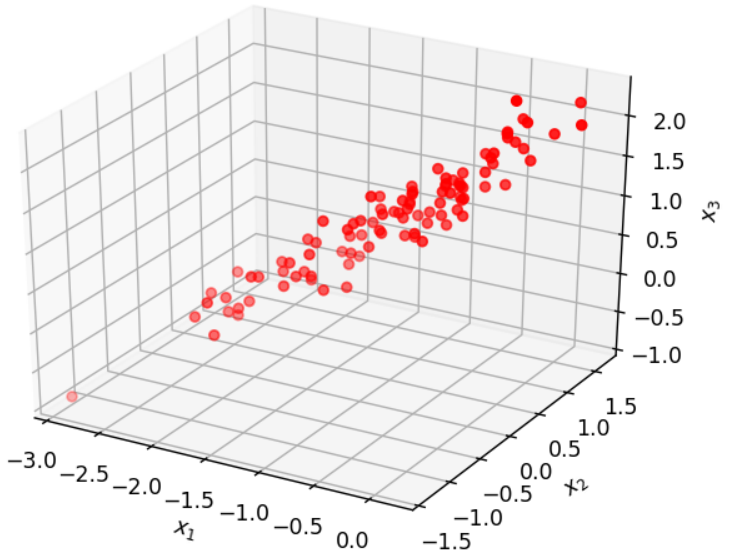
```
C:\Users\Ayan Deep Hazra\miniconda3\Lib\site-packages\numpy\lib\stride_tricks.py:116: VisibleDeprecatio
nWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples
-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify
'dtype=object' when creating the ndarray
  array = np.array(array, copy=False, subok=subok)
C:\Users\Ayan Deep Hazra\miniconda3\Lib\site-packages\numpy\core\_asarray.py:136: VisibleDeprecationWar
ning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or
ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dt
ype=object' when creating the ndarray
  return array(a, dtype, copy=False, order=order, subok=True)
C:\Users\Ayan Deep Hazra\miniconda3\Lib\site-packages\numpy\core\_asarray.py:83: VisibleDeprecationWarn
ing: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or n
darrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dty
pe=object' when creating the ndarray
  return array(a, dtype, copy=False, order=order)
```

Same process but with mean removed

In [7]:
```
# Subtract mean
X_m = X #- np.mean(X, 1).reshape((3,1))
x_m, y_m, z_m = X_m
```

In [8]:
```
# display zero mean scatter plot

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.scatter(x_m, y_m, z_m, c='r', marker='o')

ax.set_xlabel('$x_1$')
ax.set_ylabel('$x_2$')
ax.set_zlabel('$x_3$')

plt.show()
```

Figure 4



In [9]:
```
# Use SVD to find first principal component

U,s,VT = np.linalg.svd(X_m,full_matrices=False)

# complete the next line of code to assign the first principal component to a
a = U[:,[0]]

print(a)
```

```
[[-0.57725541]
 [ 0.39008946]
 [ 0.71736072]]
```

In [10]:
```
# display zero mean scatter plot and first principal component

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.scatter(x_m, y_m, z_m, c='r', marker='o', label='Data')

ax.scatter(a[0],a[1],a[2], c='c', marker='s')

ax.set_xlabel('$x_1$')
ax.set_ylabel('$x_2$')
ax.set_zlabel('$x_3$')

ax.plot([0,a[0]],[0,a[1]],[0,a[2]], c='b',label='Principal Component')

ax.legend()
plt.show()
```

Figure 5