กลุ่ม เด็กดีขี้เมา

60010113 คณิศร พิทักษ์วงศ์, 60010479 ธีระสาร มินทะขัด

ตอนที่ 2: การทดลองการลดมิติของข้อมูลด้วยค่า Principle Component Analysis

1. เตรียมชุดข้อมูล feature3 ค่า $\mathcal{X}$(accelerateX,accelerateY,accelerateZ)

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import seaborn as sns
```

In [2]:
```
df = pd.read_csv('watch_test2_sample.csv')
df['uts'] = pd.to_datetime(df['uts'])
```

In [3]:
```
df = df.drop_duplicates()
```

In [4]:
```
df = df.set_index('uts')
df = df.fillna(df.mean())
```

In [5]:
```
accelerator_df = df[['accelerateX', 'accelerateY', 'accelerateZ']].copy()
accelerator_df
```

In [1] ทำการ import library ที่จำเป็นต่อการทดลอง

In [2] อ่านไฟล์จาก .csv โดยใช้ pandas แล้วเก็บลง df แล้วแปลง format ของ feature “uts” จาก string เป็น datetime และทำการ copy ข้อมูลไว้เพื่อใช้งานในการplot GPS จากค่าจริง

In [3] ทำการลบข้อมูลที่มีค่าซ้ำซ้อน

In [4] กำหนด index เป็น feature ของ uts เพราะ ข้อมูลถูกเก็บแบบ timestamp แล้วแทนข้อมูลที่หายไป หรือ NaN ด้วย mean ของแต่ละ feature

In [5] จัด array ให้มี feature 3 คือ [accelerateX, accelerateY, accelerateZ]

Out[5]:

uts	accelerateX	accelerateY	accelerateZ
2018-11-18 08:18:41+07:00	-3.957379	-14.204506	2.303692
2018-11-18 08:19:03+07:00	-0.038236	-1.156625	1.883101
2018-11-18 08:19:45+07:00	1.906998	-4.394027	-4.358852
2018-11-18 08:20:13+07:00	-0.265259	-10.149148	3.042116
2018-11-18 08:20:33+07:00	-2.098175	-11.195846	1.754056
...	...	...	...
2018-11-18 16:07:32+07:00	4.803340	0.050184	-8.263658
2018-11-18 16:07:58+07:00	1.780343	-6.609970	2.081448
2018-11-18 16:08:19+07:00	0.399084	-8.364026	-5.357756
2018-11-18 16:08:42+07:00	0.745593	-8.820463	-5.106835
2018-11-18 16:09:04+07:00	0.480334	-8.700976	-4.669516

271 rows × 3 columns

2. ปรับให้เป็น Zero Mean และ คำนวณค่า covariance

```
In [6]: accelerator_df = accelerator_df - accelerator_df.mean()
```

```
In [7]: accelerator_df
```

Out[7]:

uts	accelerateX	accelerateY	accelerateZ
2018-11-18 08:18:41+07:00	-5.618468	-9.810479e+00	-1.596343
2018-11-18 08:19:03+07:00	-1.699325	3.237402e+00	-2.016934
2018-11-18 08:19:45+07:00	0.245909	1.776357e-15	-8.258887
2018-11-18 08:20:13+07:00	-1.926348	-5.755121e+00	-0.857919
2018-11-18 08:20:33+07:00	-3.759265	-6.801819e+00	-2.145979
...	...	...	...
2018-11-18 16:07:32+07:00	3.142251	4.444211e+00	-12.163693
2018-11-18 16:07:58+07:00	0.119253	-2.215943e+00	-1.818587
2018-11-18 16:08:19+07:00	-1.262006	-3.969999e+00	-9.257791
2018-11-18 16:08:42+07:00	-0.915496	-4.426436e+00	-9.006870
2018-11-18 16:09:04+07:00	-1.180755	-4.306949e+00	-8.569551

271 rows × 3 columns

```
In [8]: accelerator = accelerator_df.to_numpy()
```

```
In [9]: cov = accelerator.T.dot(accelerator)/(len(accelerator) - 1)
        cov
```

```
Out[9]: array([[25.4476783 ,  3.15359882, -1.82430517],
               [ 3.15359882, 23.14423757,  7.02704944],
               [-1.82430517,  7.02704944, 19.5368561 ]])
```

```
In [10]: eigen_values, eigen_vectors = np.linalg.eig(cov)
         eigen_values, eigen_vectors
```

```
Out[10]: (array([13.15046863, 25.75818909, 29.22011425]),
          array([[ 0.26701286, -0.87106272,  0.41225462],
                 [-0.60926567,  0.17885949,  0.77253131],
                 [ 0.74665888,  0.45744838,  0.48295082]]))
```

```
In [11]: sorted_indexes = np.argsort(eigen_values)[::-1]
         eigen_values = eigen_values[sorted_indexes]
         eigen_vectors = eigen_vectors[:, sorted_indexes]
         eigen_values, eigen_vectors
```

```
Out[11]: (array([29.22011425, 25.75818909, 13.15046863]),
          array([[ 0.41225462, -0.87106272,  0.26701286],
                 [ 0.77253131,  0.17885949, -0.60926567],
                 [ 0.48295082,  0.45744838,  0.74665888]]))
```

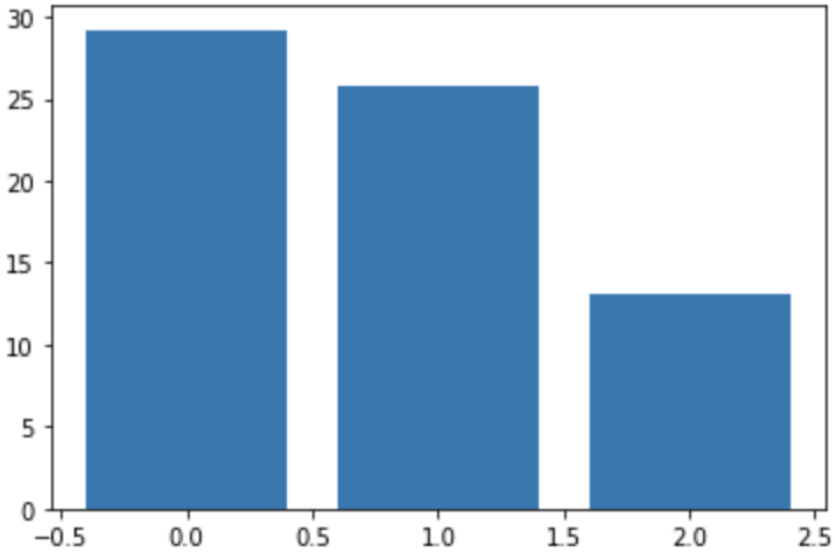
ln [9] คำนวณค่า covariance matrix ของชุดข้อมูล

ln [10] คำนวณค่า eigenvalue / eigenvector จาก covariance matrix ที่คำนวณได้จาก ln [9]

ln [11] sort array ที่ได้จาก ln [10] เพื่อเตรียม plot graph ในขั้นต่อไป

3. แสดงกราฟ Eigen Space (Eigenvalue, Eigenvector)

```
In [12]: plt.bar(np.arange(len(eigen_values)), eigen_values)
Out[12]: <BarContainer object of 3 artists>
```



In [12] แสดงกราฟแท่ง (Bar graph) ของค่า Eigenvalue ที่จัดเรียงค่าจากมากไปน้อย

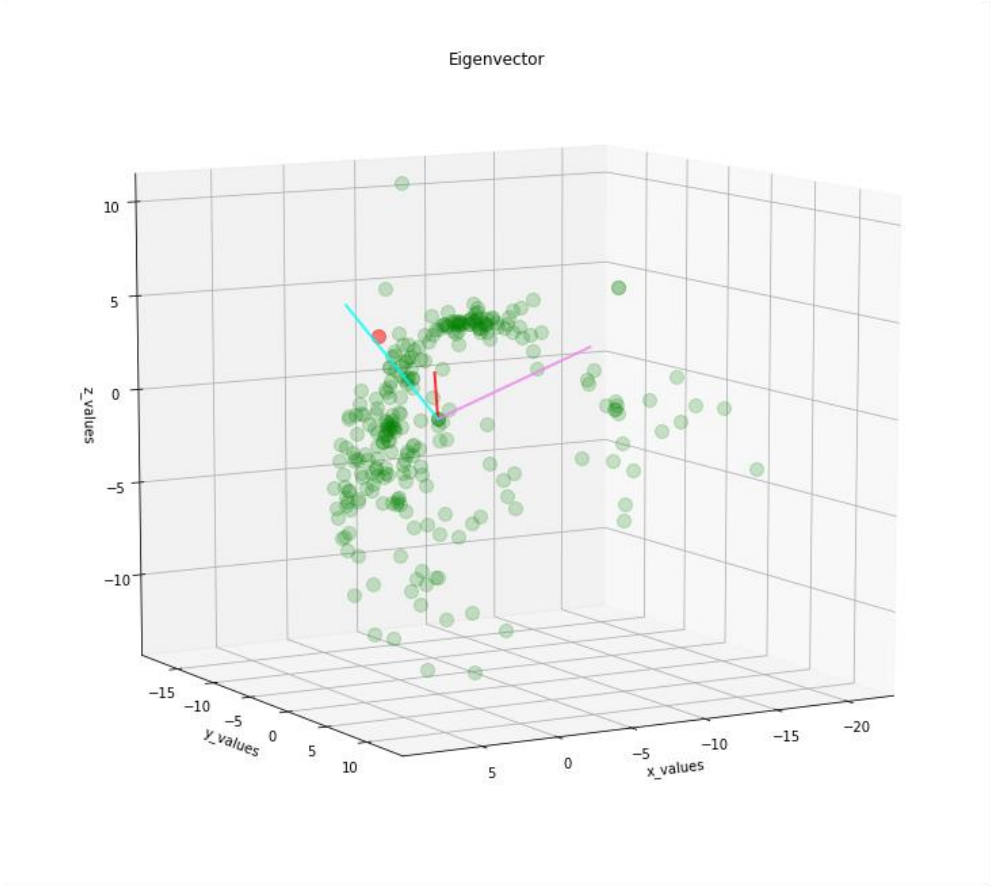
```
In [13]: np.sqrt(eigen_values)
Out[13]: array([5.40556327, 5.07525261, 3.62635749])

In [14]: eigen_vectors
Out[14]: array([[ 0.41225462, -0.87106272,  0.26701286],
                [ 0.77253131,  0.17885949, -0.60926567],
                [ 0.48295082,  0.45744838,  0.74665888]])

In [15]: eigen_vectors = (eigen_vectors.T * np.sqrt(eigen_values)).T
```

In [15] ปรับขนาดของ Eigenvector ด้วยค่า Eigenvalue จากสูตร

$$\text{Eigenvector} * \sqrt{Eigenvalue}$$

```
In [16]: scale = 2
         ev1, ev2, ev3 = eigen_vectors.T * scale
```

```
In [17]: fig = plt.figure(figsize=(50, 10))

         ax = fig.add_subplot(141, projection='3d')

         ax.plot(accelerator[:, 0],accelerator[:, 1],accelerator[:, 2],'o',markersize=10,color='green',alpha=0.2)

         ax.plot([df['accelerateX'].mean()],[df['accelerateY'].mean()],[df['accelerateZ'].mean()],'o',markersize=10,color='red'
                 ,alpha=0.5)

         ax.plot([0, ev1[0]], [0, ev1[1]], [0, ev1[2]],color='red', alpha=0.8, lw=2)
         ax.plot([0, ev2[0]], [0, ev2[1]], [0, ev2[2]],color='violet', alpha=0.8, lw=2)
         ax.plot([0, ev3[0]], [0, ev3[1]], [0, ev3[2]],color='cyan', alpha=0.8, lw=2)

         ax.set_xlabel('x_values')
         ax.set_ylabel('y_values')
         ax.set_zlabel('z_values')

         plt.title('Eigenvector')

         ax.view_init(10, 60)

         plt.show()
```



ln [16] เตรียมค่าเพื่อนำในสร้าง graph ตามสูตรด้านล่าง

ln [17] แสดงกราฟความสัมพันธ์ของ feature และ eigen vector จาก ln [16]

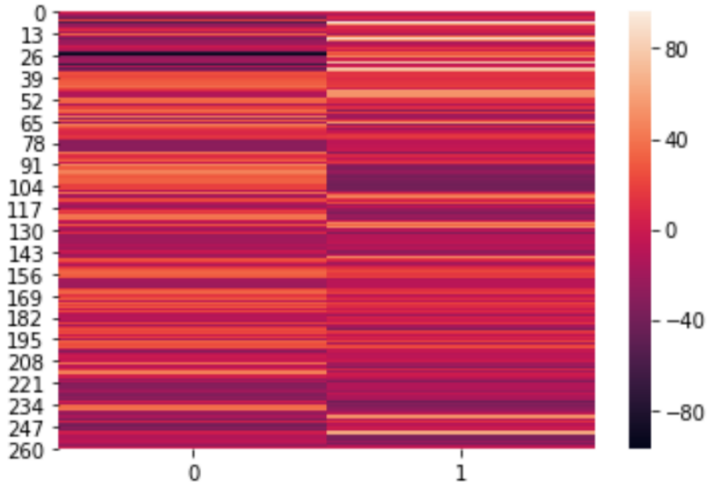
```
In [18]: K = 2
         pca = accelerator.dot( eigen_vectors[:, :K])
         pca
```

```
Out[18]: array([[-5.37811822e+01,  1.49013734e+01],
                [ 5.37392288e+00,  7.59435471e+00],
                [-1.39162216e+01, -1.48583130e+01],
                [-2.83599554e+01,  2.42294504e+00],
                [-3.88042832e+01,  7.96651355e+00],
                [-5.21090850e+01,  4.66019934e+01],
                [ 1.83454424e+01, -1.31337096e+01],
                [-5.67352295e+01,  9.60914170e+01],
```

ln [18] ลดมิติของข้อมูลจาก 3D features $\mathcal{X}$ (accelerateX, accelerateY, accelerateZ) ลงเหลือ 2D โดย เลือก eigenvector 2 vector แรก ที่สัมพันธ์กับ eigenvalue ที่มีค่าสูงสุด 2 อันดับแรก [np.dot()]

```
In [20]: sns.heatmap(pca)
```

```
Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x1e2b732d7c8>
```



ln [20] แสดง graph heatmap จากข้อมูลที่ได้ใน ln [18]