**เอกสารอธิบายการทดลองที่ 1 ตอนที่ 2**

**พื้นฐานการอ่านไฟล์ข้อมูล การแก้ปัญหาข้อมูลหาย การปรับช่วงค่าของข้อมูล   
การปรับลดมิติข้อมูล และแสดงผลข้อมูลในเชิงกราฟ**

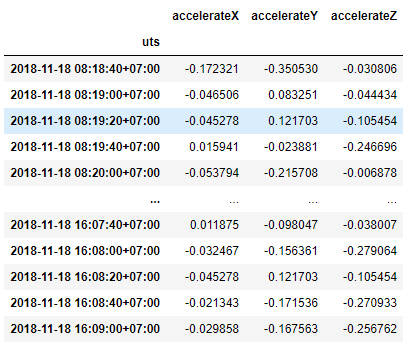
**ตอนที่ 2:** การทดลองการลดมิติของข้อมูลด้วยค่า Principle Component Analysis

* 1. เตรียมชุดข้อมูล feature 3 ค่า 𝑥(accelerateX, accelerateY, accelerate
* ปรับให้เป็น zero mean: โดยจะทำการวนแต่ละแถว ซึ่งค่าของ column นั้นๆ จะถูกลบกับค่าเฉลี่ยของ column นั้นๆ

for column in df.columns :

    df[column] = df.apply(lambda row: (row[column] - df[column].mean()), axis=1)

df



* คำนวณค่า covariance matrix ของชุดข้อมูล Xnorm จากสูตรจะได้

covariance\_matrix = np.dot(df.T, df) / (len(df) -1)

covariance\_matrix



* 1. คำนวณค่า eigenvalue / eigenvector จาก covariance matrix ที่คำนวณได้จากข้อ 2.1

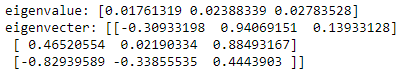
หา eigenvalue และ eigenvector จาก ฟังก์ชัน np.linalg.eig โดยจะใส่พารามิเตอร์เป็น covariance matrix ก่อนหน้า

# คำนวณค่า eigenvalue / eigenvector จาก covariance matrix

eigenvalue, eigenvecter = np.linalg.eig(covariance\_matrix)

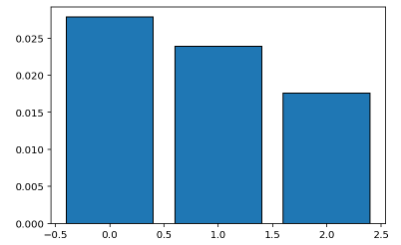
print('eigenvalue:', eigenvalue)

print('eigenvecter:', eigenvecter)



* 1. แสดงกราฟ Eigen Space (Eigenvalue, Eigenvector)
* แสดงกราฟแท่ง (Bar graph) ของค่า Eigenvalue ที่จัดเรียงค่าจากมากไปน้อย

plt.bar(np.arange(3), height=np.sort(eigenvalue)[::-1], width=0.8, align='center', edgecolor='k')



* แสดงปรับขนาดของ Eigenvector ด้วยค่า Eigenvalue

ev1 = eigenvecter[: ,0] \* np.sqrt(eigenvalue[0])

ev2 = eigenvecter[: ,1] \* np.sqrt(eigenvalue[1])

ev3 = eigenvecter[: ,2] \* np.sqrt(eigenvalue[2])

print(ev1, ev2, ev3)



* แสดงกราฟความสัมพันธ์ของ feature และ eigen vector

fig = plt.figure(figsize=(35, 8))

ax = fig.add\_subplot(141, projection='3d')

ax.plot(df['accelerateX'], df['accelerateY'], df['accelerateZ'], '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)

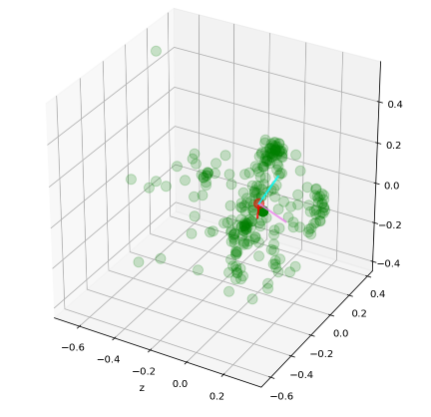
ax.set\_xlabel('x')

ax.set\_xlabel('y')

ax.set\_xlabel('z')

ax.view\_init(30, -60)

plt.show()



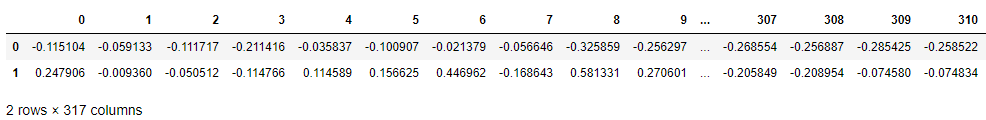
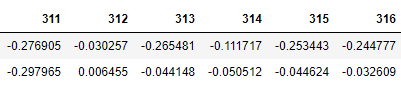
* 1. ทำ PCA เพื่อลดข้อมูลจาก 3D -> 2D

- ลดมิติของข้อมูลจาก 3D features 𝑥 (accelerateX, accelerateY, accelerateZ) ลงเหลือ 2D โดย เลือก eigenvector 2 vector แรก ที่สัมพันธ์กับ eigenvalue ที่มีค่าสูงสุด 2 อันดับแรก

x\_pca =np.dot(np.take(eigenvecter,np.argsort(eigenvalue)[1:].tolist(),axis=0),df.T)

df\_x\_pca = pd.DataFrame(x\_pca)

df\_x\_pca



- แสดงภาพ 𝑋\_𝑃𝐶𝐴 ด้วย sns.heatmap()

sns.heatmap(df\_x\_pca)

