```python
import pandas as pd
import matplotlib.pyplot as plt
import os

if not os.path.exists('./Output'):
    os.mkdir('Output')

if not os.path.exists('./Output/csv'):
    os.mkdir('./Output/csv')

if os.path.exists('MESURES.csv'):

    headerName = ["flag", "dt", "gravity_x", "gravity_y", "gravity_z", "gyro_x", "gyro_y", "gyro_z", "mag_x", "mag_y", "mag_z", "accel_x", "accel_y", "accel_z", "euler_h", "euler_p", "euler_r", "quater_w", "quater_x", "quater_y", "quater_z"]

    df = pd.read_csv('MESURES.csv', sep=';', index_col=False, names=headerName, lineterminator='\n')

    df.head()
    df.to_excel(r'./Output/Mesures.xlsx', index = False, header=True)

    df_t = df.transpose()

    flg = pd.DataFrame(df_t.loc['flag'].to_numpy().reshape(1,-1))

    dt = pd.DataFrame(df_t.loc['dt'].to_numpy().reshape(1,-1))



    gravity_x = pd.DataFrame(df_t.loc['gravity_x'].to_numpy().reshape(1,-1))
    gravity_y = pd.DataFrame(df_t.loc['gravity_y'].to_numpy().reshape(1,-1))
    gravity_z = pd.DataFrame(df_t.loc['gravity_z'].to_numpy().reshape(1,-1))

    gyro_x = pd.DataFrame(df_t.loc['gyro_x'].to_numpy().reshape(1,-1))
    gyro_y = pd.DataFrame(df_t.loc['gyro_y'].to_numpy().reshape(1,-1))
    gyro_z = pd.DataFrame(df_t.loc['gyro_z'].to_numpy().reshape(1,-1))

    mag_x = pd.DataFrame(df_t.loc['mag_x'].to_numpy().reshape(1,-1))
    mag_y = pd.DataFrame(df_t.loc['mag_y'].to_numpy().reshape(1,-1))
    mag_z = pd.DataFrame(df_t.loc['mag_z'].to_numpy().reshape(1,-1))

    accel_x = pd.DataFrame(df_t.loc['accel_x'].to_numpy().reshape(1,-1))
    accel_y = pd.DataFrame(df_t.loc['accel_y'].to_numpy().reshape(1,-1))
    accel_z = pd.DataFrame(df_t.loc['accel_z'].to_numpy().reshape(1,-1))

    euler_h = pd.DataFrame(df_t.loc['euler_h'].to_numpy().reshape(1,-1))
    euler_p = pd.DataFrame(df_t.loc['euler_p'].to_numpy().reshape(1,-1))
    euler_r = pd.DataFrame(df_t.loc['euler_r'].to_numpy().reshape(1,-1))

    quater_w = pd.DataFrame(df_t.loc['quater_w'].to_numpy().reshape(1,-1))
    quater_x = pd.DataFrame(df_t.loc['quater_x'].to_numpy().reshape(1,-1))
    quater_y = pd.DataFrame(df_t.loc['quater_y'].to_numpy().reshape(1,-1))
    quater_z = pd.DataFrame(df_t.loc['quater_z'].to_numpy().reshape(1,-1))


    flg.to_csv(r'./Output/csv/flag.csv', sep=";", index=False, header=False, line_terminator=None)
    dt.to_csv(r'./Output/csv/dt.csv', sep=";", index=False, header=False, line_terminator=None)

    gravity_x.to_csv(r'./Output/csv/gravity_x.csv', sep=";", index=False, header=False, line_terminator=None)
    gravity_y.to_csv(r'./Output/csv/gravity_y.csv', sep=";", index=False, header=False, line_terminator=None)
    gravity_z.to_csv(r'./Output/csv/gravity_z.csv', sep=";", index=False, header=False, line_terminator=None)

    gyro_x.to_csv(r'./Output/csv/gyro_x.csv', sep=";", index=False, header=False, line_terminator=None)
    gyro_y.to_csv(r'./Output/csv/gyro_y.csv', sep=";", index=False, header=False, line_terminator=None)
    gyro_z.to_csv(r'./Output/csv/gyro_z.csv', sep=";", index=False, header=False, line_terminator=None)

    mag_x.to_csv(r'./Output/csv/mag_x.csv', sep=";", index=False, header=False, line_terminator=None)
    mag_y.to_csv(r'./Output/csv/mag_y.csv', sep=";", index=False, header=False, line_terminator=None)
    mag_z.to_csv(r'./Output/csv/mag_z.csv', sep=";", index=False, header=False, line_terminator=None)

    accel_x.to_csv(r'./Output/csv/accel_x.csv', sep=";", index=False, header=False, line_terminator=None)
    accel_y.to_csv(r'./Output/csv/accel_y.csv', sep=";", index=False, header=False, line_terminator=None)
    accel_z.to_csv(r'./Output/csv/accel_z.csv', sep=";", index=False, header=False, line_terminator=None)

    euler_h.to_csv(r'./Output/csv/euler_h.csv', sep=";", index=False, header=False, line_terminator=None)
    euler_p.to_csv(r'./Output/csv/euler_p.csv', sep=";", index=False, header=False, line_terminator=None)
    euler_r.to_csv(r'./Output/csv/euler_r.csv', sep=";", index=False, header=False, line_terminator=None)

    quater_w.to_csv(r'./Output/csv/quater_w.csv', sep=";", index=False, header=False, line_terminator=None)
    quater_x.to_csv(r'./Output/csv/quater_x.csv', sep=";", index=False, header=False, line_terminator=None)
    quater_y.to_csv(r'./Output/csv/quater_y.csv', sep=";", index=False, header=False, line_terminator=None)
    quater_z.to_csv(r'./Output/csv/quater_z.csv', sep=";", index=False, header=False, line_terminator=None)


    fig, ((grav, gyro, mag), (acc, euler, quater)) = plt.subplots(2, 3)
    fig.suptitle('Mesures IMU')

    #flag.grid()
    #flag.plot(df['flag'])
    #flag.set_ylabel(r'Flag mesure')


    grav.grid()
    grav.plot(df['gravity_x'], label="X")
    grav.plot(df['gravity_y'], label="Y")
    grav.plot(df['gravity_z'], label="Z")
    grav.legend(loc=1, prop={'size': 7})
    grav.set_ylabel(r'Gravity $[m/s^2]$')

    gyro.grid()
    gyro.plot(df['gyro_x'], label="X")
    gyro.plot(df['gyro_y'], label="Y")
    gyro.plot(df['gyro_z'], label="Z")
    gyro.legend(loc=1, prop={'size': 7})
    gyro.set_ylabel('Gyroscope $[°/s]$')

    mag.grid()
    mag.plot(df['mag_x'], label="X")
    mag.plot(df['mag_y'], label="Y")
    mag.plot(df['mag_z'], label="Z")
    mag.legend(loc=1, prop={'size': 7})
    mag.set_ylabel('Magnitude $[uT]$')

    acc.grid()
    acc.plot(df['accel_x'], label="X")
    acc.plot(df['accel_y'], label="Y")
    acc.plot(df['accel_z'], label="Z")
    acc.legend(loc=1, prop={'size': 7})
    acc.set_ylabel('Accel. $[m/s]$')

    euler.grid()
    euler.plot(df['euler_h'], label="H")
    euler.plot(df['euler_p'], label="P")
    euler.plot(df['euler_r'], label="R")
    euler.legend(loc=1, prop={'size': 7})
    euler.set_ylabel('Euler')

    quater.grid()
    quater.plot(df['quater_w'], label="W")
    quater.plot(df['quater_x'], label="X")
    quater.plot(df['quater_y'], label="Y")
    quater.plot(df['quater_z'], label="Z")
    quater.legend(loc=1, prop={'size': 7})
    quater.set_ylabel('Quaternion')


    manager = plt.get_current_fig_manager()
    manager.full_screen_toggle()
    plt.savefig("./Output/MESURES_GRAPH.svg")
    plt.savefig("./Output/MESURES_GRAPH.png", dpi=300)
    plt.show()

else :
    print("Pas de fichier MESURES.csv")
```