

Kod źródłowy

```
In [ ]: import numpy as np
from read_flightradar import read_flightradar
import folium
import matplotlib.pyplot as plt

a = 6378137 # wielka półoś elipsoidy GRS80 w metrach
e2 = 0.00669438002290 # kwadrat pierwszego mimośrodu dla elipsoidy GRS80

def Rneu(phi, lamb):
    R = np.array([[ -np.sin(phi)*np.cos(lamb), -np.sin(lamb), np.cos(phi)*np.cos(lamb)],
                  [ -np.sin(phi)*np.sin(lamb), np.cos(lamb), np.cos(phi)*np.sin(lamb)],
                  [ np.cos(phi), 0, np.sin(phi) ]])
    return R

def flh2xyz(phi, lamb, h):
    N = a/np.sqrt(1-e2*np.sin(phi)**2)
    x = (N + h)*np.cos(phi)*np.cos(lamb)
    y = (N + h)*np.cos(phi)*np.sin(lamb)
    z = (N*(1-e2)+h)*np.sin(phi)
    return [x, y, z]

file = 'lot4.csv'
dane = read_flightradar(file)

flh = dane[:, [7, 8, 9]]
flh[:, -1] = flh[:, -1]*0.3048 + 135.40

flh_lotnisko = flh[0, :]
flh = flh[66:, :]

xyz_lotnisko = flh2xyz(np.deg2rad(flh_lotnisko[0]), np.deg2rad(flh_lotnisko[1]), flh_lotnisko[2])

R = Rneu(np.deg2rad(flh_lotnisko[0]), np.deg2rad(flh_lotnisko[1]))

azimuths = []
altitudes = []
distances = []
hrzaltitudes = []
locations = []

start_time = dane[0, 0]
times = []

for i in dane[:, 0]:
    times.append((i - start_time)/3600)

times = times[66:]

breakpoint = None

speeds = []
spd = dane[:, -2]
for i in spd:
    speeds.append(i)
```

```

speeds = speeds[66:]

for fi, lam, h in flh:
    xyz = flh2xyz(np.deg2rad(fi), np.deg2rad(lam), h)

    xs1 = np.array(xyz) - xyz_lotnisko
    neu = R.T @ xs1

    az = np.rad2deg(np.arctan2(neu[1], neu[0]))
    s = np.sqrt(neu[0]**2 + neu[1]**2 + neu[2]**2) / 1000
    hrzalt = np.arcsin(neu[2]/(1000*s))

    hrzaltitudes.append(hrzalt)
    azimuths.append(az)
    distances.append(s)
    altitudes.append(h)
    locations.append([fi, lam])

for i in range(len(hrzaltitudes)):
    if hrzaltitudes[i] < 0:
        breakpoint = i
        break

# mapa Lotu
map = folium.Map(location=[52.164318, 20.981787], zoom_start=6)
folium.PolyLine(locations[:i], color='green', weight=2.5, opacity=1).add_to(map)
folium.PolyLine(locations[i-1:], color='red', weight=2.5, opacity=1).add_to(map)
map.save('mapa.html')

# wykres skyplot
fig = plt.figure()
ax = fig.add_subplot(111, projection='polar')
ax.set_theta_direction(-1)
ax.set_theta_offset(np.pi/2)
ax.plot(np.deg2rad(azimuths[:i]), distances[:i], 'o', color='orange', markersize=5)
plt.savefig('skyplot.png', dpi=300)

# wykres wysokości samolotu
plt.figure()
plt.plot(times[:i], altitudes[:i], color='green')
plt.plot(times[i-1:], altitudes[i-1:], color='red')
plt.ylim(0, 12000)
plt.xlim(0, 7.5)
plt.xlabel('czas lotu[h]')
plt.ylabel('wysokość [m]')
plt.title('wysokość samolotu w zależności od czasu lotu')
plt.xticks(np.arange(0, 8, 0.5))
plt.grid()
plt.savefig('wswzoc.png', dpi=300)

# wykres prędkości samolotu
plt.figure()
plt.plot(times[:i], speeds[:i], color='green')
plt.plot(times[i-1:], speeds[i-1:], color='red')
plt.ylim(0, 500)
plt.xlim(0, 7.6)
plt.xlabel('czas lotu[h]')

```

```
plt.ylabel('prędkość [km/h]')
plt.title('prędkość samolotu w zależności od czasu lotu')
plt.xticks(np.arange(0, 8, 0.5))
plt.grid()
plt.savefig('pswzoc.png', dpi=300)

# wykres odległości samolotu od lotniska początkowego
plt.figure()
plt.plot(times[:i], distances[:i], color='green')
plt.plot(times[i-1:], distances[i-1:], color='red')
plt.ylim(0)
plt.xlim(0, 7.7)
plt.xlabel('czas lotu[h]')
plt.ylabel('odległość [km]')
plt.title('odległość samolotu od lotniska początkowego w zależności od czasu lotu')
plt.xticks(np.arange(0, 8, 0.5))
plt.grid()
plt.savefig('wzowzoc.png', dpi=300)

plt.show()
```