Dr. Jesús Martí Gavilá. Departamento de Ingeniería Cartográfica, Geodesia y fotogrametría

PRÁCTICA 7

CURSO BÁSICO PYTHON: FICHERO GPX

Grado Tecnologías Interactivas | Tecnologías de la Información Geográfica

TAREA

Realizar un análisis de rendimiento a partir de un fichero GNSS con waypoints en formato GPX.

El resultado a obtener sera:

- Un Dataframe en formato HTML con los siguientes campos:

'Punto','Latitud','Longitud', 'Cota (m)', 'Hora', 'Distancia (m)','Dist_O (m)', 'Pendiente (%)', 'Velocidad (km/h)','Orientación (º)','Tiempo'

- Gráfico con los valores de distancia en el eje X y valores de pendiente y velocidad en la Y
- Mapa interactivo

Estructura GPX e importación de librerías

In [ ]:
```
"""
Creado Viernes 14 Oct 18:22:39 2022, @autor/es: ..............................
FORMATO EJEMPLO GPX
<wpt lat="38.968484" lon="-0.148688">
        <ele>-6.847729</ele>
        <time>2020-10-06T15:42:19Z</time>
        <name>001</name>
        <sym>Flag, Blue</sym>
```

```
</wpt>
"""
import pandas as pd
import xml.etree.cElementTree as et
import os
import math as m
from datetime import datetime
import matplotlib.pyplot as plt
```

Fichero de entrada y Dataframe de salida

In [ ]:
```
# Introducción de datos. Solicitar al usuario el nombre del fichero sin extensió
.....................
.....................
# Este nombre también se utilizará para crear el fichero html
.....................
.....................
```

In [ ]:
```
# Accedemos al XML y creamos un Dataframe con los campos arriba indicados
docxml = et.parse(pathgpx)
raiz = docxml.getroot()
puntos = len(raiz)-1
columnas = ['Punto','Latitud','Longitud', 'Cota (m)', 'Hora', 'Distancia (m)','D
df = pd.DataFrame(columns=columnas)
```

Funciones de cálculo

In [ ]:
```
# Cálculo de la distancia parcial . función distacias
def distancias ..............................
```

In [ ]:
```
# Cálculo del acimut. Verificar la posición del punto B. funcion orientación
def orientacion ..............................
```

In [ ]:
```
# Cálculo de la pendiente. En % y angulo. funcion pendiente
def pendiente ..............................
```

In [ ]:
```
# Cálculo de la velocidad en km/h. funcion velocidad
def velocidad ..............................
```

Obtención de los datos de entrada y llamada a funciones

In [ ]:
```
waypoints = range (1,puntos,1)
n = 0.00
for i in waypoints:
    # Datos del punto A
    wpt1 = raiz[i]
    lat1_g = float(wpt1.get("lat"))
    lon1_g = float(wpt1.get("lon"))
    lat1 = m.radians (lat1_g)
    lon1 = m.radians (lon1_g)
```

```
    ele1 = round(float(wpt1[0].text),3)
    time1 = wpt1[1].text
    if i == 1:
        df_1 = pd.DataFrame([[i,lat1_g,lon1_g,ele1,time1,n,n,n,n,n,n]],columns =
        df = pd.concat([df if not df.empty else None,df_1],ignore_index=True)  #
# Incrementamos puntos
    i = i+1

# Datos del punto B
    wpt2 = raiz[i]
    lat2_g = float(wpt2.get("lat"))
    lon2_g = float(wpt2.get("lon"))
    lat2 = m.radians(lat2_g)
    lon2 = m.radians(lon2_g)
    ele2 = round(float(wpt2[0].text),3)
    time2 = wpt2[1].text

# Llamar a las diferentes funciones y añadir los datos resultantes al Dataframe
    d, dr = distancias(.............................) #Introducir los parámetros
    acimut = orientacion(.............................) #Introducir los parámetros
    peng, pend, dg = pendiente(.....................) #Introducir los parámetro
    tiempo, velkmh = velocidad(.....................)#Introducir los parámetros
    df_1 = pd.DataFrame([[i,lat2_g,lon2_g,ele2,time2,dg,0.00,pend,velkmh,acimut,
    df = pd.concat([df,df_1],ignore_index=True )

    df['Dist_O (m)'] = ................................ # crear un comando par
```

Impresión y exportación de los obtenidos

In [ ]:
```
# Imprimir los valores en formato HTML
.................................................
```

In [ ]:
```
# Plotear el gráfico de rendimiento y guardar en JPG. Puedes ver ayuda en intern
# Empieza para plotear 2 ejes
fig, ax = plt.subplots()
plt.plot(df["Dist_O (m)"], df['Pendiente (%)'], marker = "o", label="Pendiente %
.....................................................................................
# indicar leyendas, titulo, etiquetas del eje X
.....................................................................................
# Exportar en JPG
.....................................................................................
```

Mapa interactivo con Folium. Código ejemplo

In [ ]:
```
#Instalar primeramente la libreria folium
.....................................................................................

#Código creación fichero decoordenadas
df.loc[:,['Latitud','Longitud']].to_csv('coordenadas.csv',header=True, index=Fal
lista_geo = pd.read_csv('coordenadas.csv').to_numpy().tolist()

#Creación del mapa interactivo
import folium
```

```
from folium import plugins
m = folium.Map()
folium.plugins.AntPath(locations = lista_geo, popup = 'Ruta 20/11/2023', reverse
m.fit_bounds(m.get_bounds())
m
```

In [ ]: