

ขั้นตอนการ แปลงไฟล์ nc ไปเป็น csv (exel)

### ขั้นที่ 3 การ แปลงไฟล์ nc



#### 3.1 เริ่มต้น import Lib ดังนี้

```
import cdsapi
```

```
import netCDF4
```

```
from netCDF4 import num2date
```

```
import numpy as np
```

```
import os
```

```
import pandas as pd
```

```
import shutil
```

```
import pathlib
```

### 3.2 เปิดไฟล์ nc ด้วยคำสั่ง

netCDF4.Dataset(file\_location)

ตัวอย่าง `f = netCDF4.Dataset("../ncData/New folder/ChiangRai_Muang.nc")`

### ขั้นที่ 4 การดูชื่อตัวแปรแต่ละตัว ที่ได้ Download มา

ใช้คำสั่ง `print(f.variables)`

รายละเอียดของตัวแปร จะอยู่ในรูปแบบนี้

```
unlimited dimensions:      -      -
current shape = (1092, 2, 1, 1)
filling on, 'sp': <class 'netCDF4._netCDF4.Variable'>
int16 sp(time, expver, latitude, longitude)
    scale_factor: 0.03117322570308089
    add_offset: 94516.54691338715
    _FillValue: -32767
    missing_value: -32767
    units: Pa
    long_name: Surface pressure
    standard name: surface air pressure
```

โดยเราจะใช้ชื่อตัวย่อของตัวแปร ในการระบุ

ให้นำข้อมูลมาเก็บไว้ในตัวแปร โดยใช้ชื่อตัวย่อในการระบุ

```
# Extract variable
u10 = f.variables['u10']
v10 = f.variables['v10']
d2m = f.variables['d2m']
t2m = f.variables['t2m']
uvb = f.variables['uvb']
e = f.variables['e']
stl1 = f.variables['stl1']
sp = f.variables['sp']
tp = f.variables['tp']
```

`sp = f.variables['sp']`

ขั้นที่ 5 เตรียม Data time, lat, lng โดยอิงจากตัวแปรที่มี (ตัวอย่างนี้ใช้ t2m)

```
# Get dimensions assuming 3D: time, latitude, longitude
time_dim, lat_dim, lon_dim = t2m.get_dims()
time_var = f.variables[time_dim.name]
times = num2date(time_var[:], time_var.units)
latitudes = f.variables[lat_dim.name][:]
longitudes = f.variables[lon_dim.name][:]
```

ในบางกรณี อาจต้องใช้

```
time_dim, expver, lat_dim, lon_dim = t2m.get_dims()
```

หาก get\_dims() Return ออกมา 4 ค่า

## ขั้นที่ 6 ใช้ Code ในการแปลง nc ไปเป็น csv ดังนี้

```
filename = os.path.join(output_dir, r'./out/'+filename+'.csv') ##สำหรับ Path ที่ใช้ Save
print(f'Writing data in tabular form to {filename} (this may take some time)...') ##บอกสแตตัสการทำงาน
times_grid, latitudes_grid, longitudes_grid = [
    x.flatten() for x in np.meshgrid(times, latitudes, longitudes, indexing='ij')]

df = pd.DataFrame({
    'time': [t.isoformat() for t in times_grid],
    'latitude': latitudes_grid,
    'longitude': longitudes_grid,
    'u10': u10[:].flatten(),
    'v10': v10[:].flatten(),
    'd2m': d2m[:].flatten(),
    't2m': t2m[:].flatten(),
    'uvb': uvb[:].flatten(),
    'e': e[:].flatten(),
    'stl1': stl1[:].flatten(),
    'sp': sp[:].flatten(),
    'tp': tp[:].flatten(),
})
df.to_csv(filename, index=False)
print('Done')
```

```
filename = os.path.join(output_dir, r'./out/'+filename+'.csv')

print(f'Writing data in tabular form to {filename} (this may take some time)...')

times_grid, latitudes_grid, longitudes_grid = [
    x.flatten() for x in np.meshgrid(times, latitudes, longitudes, indexing='ij')]

df = pd.DataFrame({
    'time': [t.isoformat() for t in times_grid],
    'latitude': latitudes_grid,
    'longitude': longitudes_grid,
    'u10': u10[:].flatten(),
    'v10': v10[:].flatten(),
    'd2m': d2m[:].flatten(),
    't2m': t2m[:].flatten(),
    'uvb': uvb[:].flatten(),
    'e': e[:].flatten(),
    'stl1': stl1[:].flatten(),
    'sp': sp[:].flatten(),
    'tp': tp[:].flatten(),
})

df.to_csv(filename, index=False)

print('Done')
```

ในบางกรณีที่ไม่สามารถทำ DataFrame ได้ โดยเกิดจาก Array ไม่เท่ากัน

ให้เช็คและเคลียค่าที่เป็น None ออกให้หมด

\*\*\*การใช้ฟังก์ชัน filter อาจตัดค่าที่เป็น 0 ออก จึงต้องใช้ loop แทน

```
u10 = list(filter(None, u10[:].flatten()))
v10 = list(filter(None, v10[:].flatten()))
d2m = list(filter(None, d2m[:].flatten()))
t2m = list(filter(None, t2m[:].flatten()))
# uvb = list(filter(None, uvb[:].flatten()))
e = list(filter(None, e[:].flatten()))
stl1 = list(filter(None, stl1[:].flatten()))
sp = list(filter(None, sp[:].flatten()))
# tp = list(filter(None, tp[:].flatten()))

tp2 = []
for ele in tp[:].flatten():
    if ele != None:
        tp2.append(ele)

uvb2 = []
for ele in uvb[:].flatten():
    if ele != None:
        uvb2.append(ele)

df = pd.DataFrame({
    'time': [t.isoformat() for t in times_grid],
    'latitude': latitudes_grid,
    'longitude': longitudes_grid,
    'u10': u10,
    'v10': v10,
    'd2m': d2m,
    't2m': t2m,
    'uvb': uvb2,
    'e': e,
    'stl1': stl1,
    'sp': sp,
    'tp': tp2,
})
```

ยังไม่แน่ใจว่าที่ Data ไม่เท่ากัน เกิดจากอะไร

แต่จากที่เคยโหลดทั้งหมด จะมีปัญหานี้แค่ในปีล่าสุด ( 2022 )