NetCDF Data Analysis and Visualization in Python

1. Introduction to NetCDF

NetCDF (Network Common Data Form) is a file format designed to store scientific data, particularly multidimensional data such as time, space, and variables like temperature or salinity.

Network Common Data Form: صيغة شبكة المعلومات العامة.
تشير إلى تنسيق بيانات مشترك يُستخدم لتبادل المعلومات العلمية عبر الشبكات

Importance of NetCDF:

NetCDF is widely used in oceanography, meteorology, climate studies, and engineering fields. It allows efficient storage and retrieval of data, making it ideal for scientific analysis.

2. Libraries Used

Python provides several libraries to work with NetCDF data effectively:

- `**xarray**`: A powerful library for processing and analyzing **multidimensional** scientific data.
- `**matplotlib**`: To plot charts.
- `**numpy**`: Used for mathematical operations on arrays.
- `**pandas**`: Provides DataFrame structures for data analysis and allows exporting to **CSV** and **Excel**.
- `**DataFrame**`**:** Works excellently with two-dimensional data (rows and columns) and is part of the Pandas library.
-`**datetime**`: For managing dates and times.

3. Reading NetCDF Files with xarray

Use the `xarray` library to read NetCDF files from the internet or local storage.

Example:

```
import xarray as xr

import pandas as pd# للتعامل مع ملفات الاكسل و غيرها من ملفات البيانات
```

```
netcdf_file = 'https://opendap1.nodc.no/opendap/physics/point/cruise/nansen_legacy-single_profile/NMDC_Nansen-Legacy_PR_CT_58US_2021708/CTD_station_P1_NLEG01-1_-_Nansen_Legacy_Cruise_-_2021_Joint_Cruise_2-1.nc'

xrds = xr.open_dataset(netcdf_file)
print(xrds)
```

To extract a specific variable such as temperature:

```
myarray = xrds['TEMP'].values

print(myarray)
```

To display details about the variable: dimensions (such as temporal or geographical dimensions) and the index:

```
print(xrds['TEMP'])
```

To display details of all variables: dimensions (such as temporal or geographical dimensions) and the index:

```
print(xrds)
```

4. Converting Data to DataFrames

To convert a specific variable into a DataFrame:

```
df = xrds['PSAL'].to_dataframe()
print(df)
```

يتم تخزين المتغيرات في **DataFrame** داخل المتغير `df` و ذلك ليتم معالجة وتحليل البيانات مثل الفلاتر، الحسابات، التلخيص، والحفظ في تنسيقات مختلفة.

To get multiple variables as a DataFrame:

```
df_multi = xrds[['TEMP', 'PSAL', 'DENS', 'SVEL']].to_dataframe()
print(df_multi)
```

5. Saving Data to External Files

To save the data as a CSV file:

```
dt = xrds.to_dataframe()
dt.to_csv('/home/alaa/Desktop/net.csv')
```

بعد تحويل البيانات إلى **DataFrame** (المخزنة في المتغير `dt`)، يتم استخدام دالة `to_csv()` لحفظ البيانات في ملف **CSV** ضمن الامتداد المطلوب. `to_csv()` هي دالة في **Pandas** تقوم بتخزين البيانات في **CSV**، وهو تنسيق شائع لتخزين البيانات في ملف نصي يمكن قراءته واستخدامه بسهولة.

dt.to_excel('/path/to/ctd_data.xlsx') لخزن البيانات على ملف اكسل.

To save the data as an Excel file:

```
dt.to_excel('/home/alaa/Desktop/net.xlsx')
```

- **TEMP**: Temperature.
- **PSAL**: Salinity.
- **DENS**: Density.
- **SVEL**: Speed Velocity.

6. Calculating the Average of a Specific Variable

Suppose we have the temperature variable 'TEMP' and want to calculate its average.

متى تستخدم كل منهما؟

Excel: عندما تحتاج إلى استخدام ميزات متقدمة مثل الرسوم البيانية، الصيغ، العمليات الرياضية كالجمع و الطرح و المعدل الخ أو التنسيقات الاخر كتغيير الالوان.

CSV: عندما تحتاج إلى تبادل البيانات بسهولة بين الأنظمة أو عند التعامل مع كميات كبيرة من البيانات بشكل بسيط.

```
average_temp = xrds['TEMP'].mean().item()
print(f 'Average Temperature: {average_temp}')
```

إذا كان TEMPيحتوي على قيم مثل [10, 15, 20, 25]،
فإن:

mean() : تحسب (10+15+20+25)/4=17.5

item(): دالة تحول الناتج من numpy.float64 إلى عدد عادي مثل float.

البيانات من نوع NumPy (numpy.float64) قد لا تكون متوافقة بشكل كامل مع كل مكتبات Python (مثلJSON ، قواعد البيانات، أو أدوات لا تدعمNumPy).

عند العمل خارج بيئة NumPy أو Xarray (مثل إدخال البيانات في تقارير، أو تصديرها كـCSV/Excel) ، تحويل النوع إلى floatباستخدام item()يجعل البيانات أكثر **وضوحًا ومرونة**.

Numpy.float64: يشبه شخصًا يحمل بطاقة هوية (ID) مع كل تفاصيله . **بينما** `float` يشبه شخصًا يحمل اسمه فقط.

print(): يعرض النتيجة على شكل:

f: يكون مفيدًا جدًا عندما يتم دمج **نصوص** مع **قيم رقمية للمتغيرات**، سواء كانت أرقامًا أو نصوصًا أو حتى تعبيرات رياضية، داخل سطر واحد من النص.

7. Plotting Data Using Matplotlib

```
import xarray as xr

import matplotlib.pyplot as plt

# Variable extract

temperature = xrds['TEMP'].values # درجة الحرارة

salinity = xrds['PSAL'].values     # الملوحة

density = xrds['DENS'].values      # الكثافة

plt.figure(figsize=(10, 6))

sc = plt.scatter(salinity, density, c=temperature, cmap='viridis', s=50)

plt.title('Density vs Salinity with Temperature', fontsize=14)

plt.xlabel('Salinity (PSU)')

plt.ylabel('Density (kg/m³)')

cbar = plt.colorbar(sc)
cbar.set_label('Temperature (°C)')
plt.grid()
plt.show()
```

. plt.scatter-

- **وظيفة scatter** تستخدم لإنشاء مخطط نقطي ثنائي الأبعاد.(2D scatter plot)
- يعتمد المخطط على إحداثيين:
  - المحور السيني (X-axis): يمثل**salinity** (الملوحة).
  - المحور الصادي (Y-axis): يمثل**densitv** (الكثافة).

-المعاملات:

1. **salinity و density:**
   - القيم على المحورين الأفقي والرأسي (X و Y) للنقاط.
2. **c=temperature:**
   - **لون النقاط** يعتمد على القيم في المتغير temperature (درجة الحرارة).
   - القيم الأكبر أو الأصغر في temperature ستؤدي إلى تغيير لون النقاط وفقًا لنظام الألوان(colormap).

cmap='viridis'-

- **خريطة الألوان (Colormap)** المستخدمة لتلوين النقاط.
- viridisهي خريطة ألوان متدرجة من الأزرق الداكن إلى الأصفر، وهي مناسبة لعرض التغيرات العددية.

`s=50` : **حجم النقاط** في المخطط (50 بكسل لكل نقطة) .يمكن استخدامه لاحقًا لإضافة شريط ألوان (Colorbar) أو إجراء تعديلات إضافية.

sc:

- يحتوي على الكائن الناتج من المخطط.(Scatter plot object)
- يمكن استخدامه لاحقًا لإضافة شريط ألوان (Colorbar) أو إجراء تعديلات إضافية.

plt.colorbar(sc):

- تُنشئ الكود شريط ألوان بناءً على الكائن (sc)مخطط النقاط الذي يحتوي على قيم الألوان.

- يربط القيم العددية (مثل درجة الحرارة) بالألوان المستخدمة في النقاط على الرسم.

cbar:

- هو كائن شريط الألوان(Colorbar object) ، يُخزن لتعديل خصائص شريط الألوان لاحقًا (مثل إضافة تسميات أو تغيير الشكل).
الكائن `sc` في الكود الذي يستخدم `plt.scatter`
يُعتبر **الكائن الذي يمثل الرسم البياني** (scatter plot) الذي تم إنشاؤه. هو ببساطة **المرجع** للمخطط النقطي (scatter plot)الذي يحتوي على المعلومات المتعلقة بالنقاط والألوان في الرسم.

```
alaa@alaa-VirtualBox:~/Desktop$ python3 lion3.py
<xarray.Dataset> Size: 44kB
Dimensions:         (PRES: 320)
Coordinates:
  * PRES            (PRES) float32 1kB 1.0 2.0 3.0 4.0 ... 318.0 319.0 320.0
Data variables: (12/33)
    PRES_QC         (PRES) float32 1kB ...
    TEMP            (PRES) float32 1kB ...
    PSAL            (PRES) float32 1kB ...
    FLU2            (PRES) float32 1kB ...
    CNDC            (PRES) float32 1kB ...
    DENS            (PRES) float32 1kB ...
    ...              ...
    OXYOCPVL-1_QC   (PRES) float32 1kB ...
    SPAR_QC         (PRES) float32 1kB ...
    PAR_QC          (PRES) float32 1kB ...
    PSAL-2_QC       (PRES) float32 1kB ...
    TEMP-2_QC       (PRES) float32 1kB ...
    ATTNZS01_QC     (PRES) float32 1kB ...
Attributes: (12/73)
    qc_manual:                      Recommendations for in-situ data Near Re...
    contact:                        datahjelp@hi.no
    distribution_statement:         These data are public and free of charge...
    naming_authority:               no.unis
    license:                        https://creativecommons.org/licenses/by/...
    data_assembly_center:           IMR
    ...                             ...
    station_name:                   P1 (NLEG01)
    _NCProperties:                  version=2,netcdf=4.6.3,hdf5=1.10.5
    date_created:                   2022-08-08T12:44:34Z
    doi:                            10.21335/NMDC-2085836005-P1_NLEG01-1
    title:                          CTD_station_P1_NLEG01-1_-_Nansen_Legacy_...
    metadata_link:                  https://doi.org/10.21335/NMDC-2085836005...
```

```
xrds = xr.open_dataset(netcdf_file)
print(xrds)
```

```
alaa@alaa-VirtualBox:~/Desktop$ python3 lion3.py
[3.735 3.738 3.739 3.741 3.736 3.737 3.736 3.742 3.736 3.738 3.783 3.833
 3.838 3.837 3.83  3.806 3.792 3.79  3                                 817
 3.814 3.812 3.792 3.709 3.704 3.642 3                                 297
 3.165 3.106 3.094 3.094 3.102 3.079 3                                 981
 2.954 2.933 2.903 2.874 2.85  2.824 2                                 705
 2.701 2.696 2.682 2.665 2.651 2.647 2.642 2.635 2.624 2.588 2.604 2.608
 2.593 2.507 2.472 2.468 2.46  2.453 2.441 2.43  2.431 2.43  2.431 2.433
 2.43  2.417 2.394 2.382 2.361 2.335 2.32  2.303 2.275 2.267 2.263 2.277
 2.27  2.265 2.262 2.256 2.241 2.216 2.201 2.191 2.176 2.135 2.11  2.091
 2.071 2.064 2.103 2.106 2.099 2.095 2.092 2.088 2.082 2.085 2.093 2.09
 2.089 2.085 2.054 2.022 2.019 2.021 2.026 2.027 2.036 2.041 2.06  2.082
```

```
myarray = xrds['TEMP'].values

print(myarray)
```

```
alaa@alaa-VirtualBox:~/Desktop$ python3 lion3.py
        TEMP
PRES
1.0     3.735
2.0     3.738
3.0     3.739
4.0     3.741
5.0     3.736
...       ...
316.0   1.287
317.0   1.287
318.0   1.287
319.0   1.287
320.0   1.287

[320 rows x 1 columns]
```

```
df = xrds['TEMP'].to_dataframe( )
print(df)
```

1.تعريف مجموعة البيانات(Dataset)

الحجم :(Size) حجم البيانات هو **44كيلوبايت**.

الأبعاد(Dimensions): يوجد بُعد واحد يسمى **(PRES)**غالبًا ما يشير إلى الضغط, ويتضمن 320 قيمة.

الإحداثيات (Coordinates) :البُعد **PRES**يحتوي على قيم من النوع float32 تبدأ من 1.0 إلى 320.0.

2 .المتغيرات (Data Variables) :

يتم عرض 12 من أصل 33 متغيرًا، وكل متغير يعتمد على الإحداثيات .**PRES**بعض المتغيرات المعروضة تشمل:

- **PRES_QC** يشير غالبًا إلى جودة بيانات الضغط.
- **TEMP :**يشير إلى بيانات درجة الحرارة.
- **PSAL:** يشير إلى الملوحة.(salinity)
- **FLU2** : يمكن أن تكون بيانات متعلقة بالفلوريسينس أو مشابه.
- **CNDC :** قد تمثل الموصلية الكهربائية.
- DENS : تمثل الكثافة.
- متغيرات أخرى مثل **OXYOCPVL-1_QC, SPAR_QC, PAR_QC** وغيرها و التي تعني الاتي:

OXYOCPVL-1_QC:جودة قياسات الأكسجين المذاب في الماء.

SPAR_QC: جودة قياسات الإشعاع النشط فوق سطح الماء.

PAR_QC:جودة قياسات الإشعاع النشط لعملية البناء الضوئي.

QC: يشير دائمًا إلى تقييم جودة البيانات (Quality Control).

3 .السمات (Attributes) :

هناك 12 سمة معروضة من أصل 73. السمات تضيف وصفًا أو توثيقًا للبيانات تتضمن:

qc_manual: إرشادات لتحليل بيانات الميدان.

contact: البريد الإلكتروني لجهة الاتصال (datahjelp@hi.no).

distribution_statement: تصريح بأن البيانات عامة ومجانية.

naming_authority: تشير إلى الجهة المسؤولة عن التسمية.

license: رابط الرخصة المطبقة على البيانات (Creative Commons).

data_assembly_center: المركز الذي جمع البيانات (IMR)

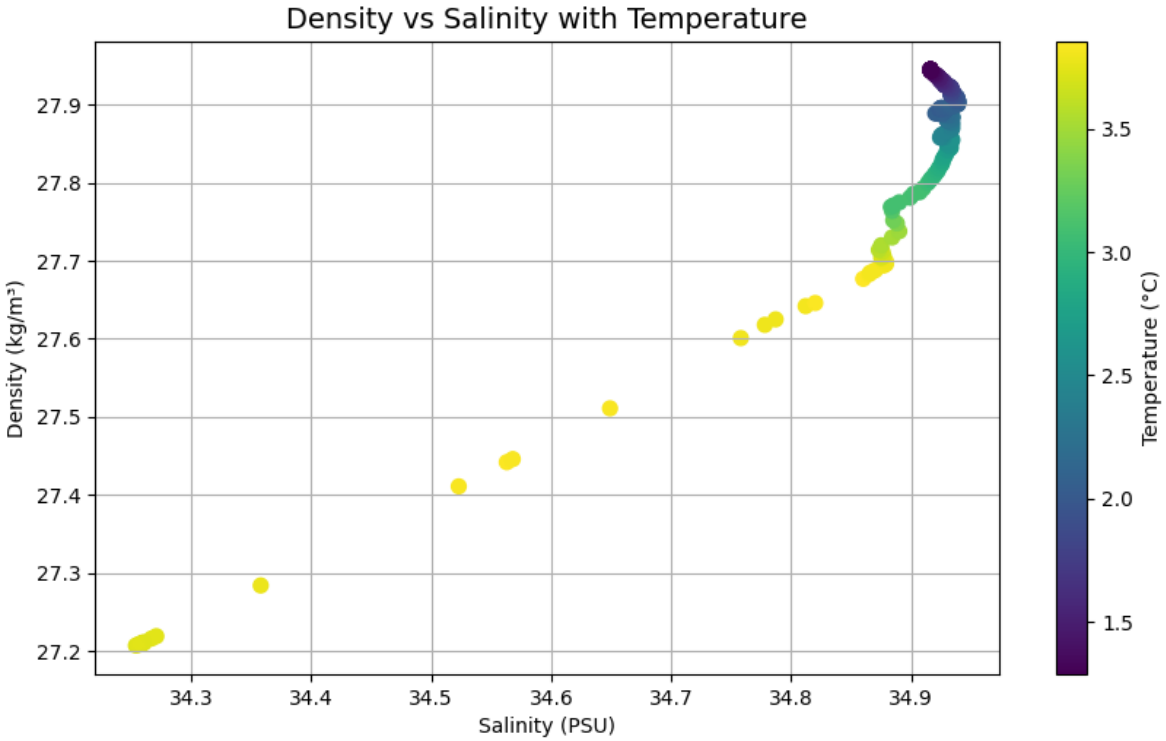
station_name: اسم المحطة (P1 - NLEG01)

date_created: تاريخ إنشاء البيانات (2022-08-08)

doi: رابط المعرف الرقمي (DOI) الخاص بالبيانات.

4 .الاستخدام العملي لهذه البيانات:

- **التحليل العلمي** :هذه البيانات يمكن استخدامها لدراسة المحيطات، مثل قياس الملوحة، درجة الحرارة، الموصلية، وغيرها.
- **الجودة (QC):** بعض المتغيرات تشير إلى تقييم جودة البيانات، مما يساعد في تحديد مدى دقة وصلاحية القياسات.
- **التوثيق** :السمات توضح مصدر البيانات، الرخصة، والاتصال بالجهة المسؤولة.


Density vs Salinity with Temperature
Density (kg/m³)
27.9
27.8
27.7
27.6
27.5
27.4
27.3
27.2
34.3
34.4
34.5
34.6
34.7
34.8
34.9
Salinity (PSU)
Temperature (°C)
3.5
3.0
2.5
2.0
1.5