Session Pratique : Analyse Comparative des Données de Précipitations Observées et de Réanalyse ERA5 avec Python#

Introduction Dans cette session, nous allons explorer comment utiliser Python et ses bibliothèques de traitement de données (pandas, numpy, xarray, matplotlib, et cartopy) pour analyser les précipitations observées. Nous comparerons ces données à celles issues de la réanalyse ERA5, fournie par le Centre Européen pour les Prévisions Météorologiques à Moyen Terme (ECMWF). Cette comparaison nous aidera à évaluer la précision des modèles de réanalyse dans un contexte local, en particulier pour les données de l'Agence Nationale de la Météorologie du Burkina Faso (ANAM).

Contexte : La Réanalyse ERA5 et les Données d'observation Les réanalyses comme ERA5 sont des reconstructions historiques de données atmosphériques, produites en utilisant des observations passées et des modèles numériques. Ces réanalyses sont cruciales pour étudier le climat, surtout dans des régions où les observations directes sont limitées. Contrairement aux données d'observation directe, ERA5 fournit des données cohérentes sur de grandes échelles, mais il est important de vérifier leur précision dans des contextes locaux. Elles sont essentielles pour combler les lacunes dans les observations, surtout dans les régions où les stations météorologiques sont peu nombreuses. Cependant, les réanalyses ne remplacent pas les données d'observation locales et peuvent varier en précision selon les régions et périodes.

Point clé : Utiliser des données de réanalyse et d’observation conjointement permet une meilleure compréhension des phénomènes climatiques locaux, mais il est crucial de connaître les limitations et d'évaluer les écarts pour les applications locales.

Les objectifs de la session: 1- Accéder et comprendre les données de réanalyse ERA5 Utiliser xarray pour charger et explorer les fichiers NetCDF de précipitations ERA5. 2- Préparer les données d'observation des stations Nettoyer et organiser les données avec pandas pour faciliter la comparaison avec ERA5. 3- Calculer des statistiques de précipitations journalières Calculer les cumuls, moyennes et écarts-types des précipitations pour chaque station pour mieux comprendre leur variabilité. 4- Visualiser et comparer les données Utiliser matplotlib et cartopy pour tracer les cumuls de précipitations, en superposant les données d'observation et de réanalyse. 5- Analyser les écarts et identifier les causes potentielles Quantifier les différences et réfléchir à leurs causes (ex. variabilité saisonnière, biais spatial).

Étape 1 : Nous allons utiliser des données journalières de précipitation observées par le réseau de stations de l’ANAM (STN_24101.csv). Voici les étapes pour charger le fichier, examiner sa structure

In [13]:
```
import pandas as pd
data = pd.read_csv('STN_24101.csv', delimiter=';')
```

Identifiez les colonnes et types de données dans le fichier. Transposer le dataframe et examiner sa nouvelle structure

In [14]:
```
# Transpose the data
transposed_data = data.transpose()

# Display the transposed data
transposed_data.head()
# Drop the first line from the transposed data
transposed_data_cleaned = transposed_data.drop(index='STN')
transposed_data.head()
```

Out[14]:

	0	1	2	3	4	5	6	7	8	9	...	185	186	
STN	Lon	Lat	20240401	20240402	20240403	20240404	20240405	20240406	20240407	20240408	...	20241001	20241002	2024
Barsalgho	-1,06	13,42	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	...	0,0	0,0	
Batie	-2,92	9,88	0,0	0,0	0,0	7,4	0,0	40,8	0,0	0,0	...	0,8	0,0	
Bereba	-3,68	11,62	0,0	0,0	0,0	0,0	12,0	0,0	0,0	0,0	...	0,0	0,0	
Bousse	-1,90	12,66	0,0	0,0	0,0	0,0	21,6	0,0	0,0	0,0	...	0,0	0,0	

5 rows × 195 columns

Calculer le cumul de précipitations pour chaque station¶. Définissez une fonction Python pour classer les stations en fonction de leurs précipitations cumulées. Pour chaque station, calculez le cumul des précipitations journalières et visualisez les résultats. Le cumul nous aide à identifier les régions à fortes précipitations sur la période étudiée. La fonction classify_climatic_zone permet de classer les stations par zone climatic. Appliquer la fonction ***apply*** *de pandas à cette nouvelle fonction pour classifier les différentes stations*

In [15]:
```
def classify_climatic_zone(precipitation):
    if precipitation < 600:
```

```
        return 'Sahelienne'
    elif precipitation < 1200:
        return 'Soudanienne'
    else:
        return 'Guinéenne'
```

En vous adaptant ce bout de code, vizualiser les cumul de précipitations de la période

Données pour les stations (remplacez-les par vos données réelles) station_names = ["Station A", "Station B", "Station C", "Station D"] Latitude = [12.4, 13.2, 10.9, 14.0] Longitude = [-1.5, -0.5, -3.5, 1.0] Pluie = [200, 450, 100, 300] # Cumul des précipitations en mm pour la période # 1. Ajuster les tailles des cercles en fonction des précipitations (complétez cette ligne) circle_sizes = _____ # Ajustez la division pour des tailles proportionnées # 2. Création de la figure et ajout de la projection fig, ax = plt.subplots(figsize=(10, 8), subplot_kw={'projection': ccrs.____()}) # 3. Ajouter les frontières et la côte (complétez ces lignes) ax.add_feature(cfeature.____, linestyle='-', alpha=0.5) ax.add_feature(cfeature.____, linestyle=':', alpha=0.5) # Centrer la carte sur votre région cible ax.set_extent([-5, 5, 9, 16], crs=ccrs.PlateCarree()) # 4. Tracer les cumuls de précipitations (complétez cette ligne) scatter = ax.scatter(____, ____, s=circle_sizes, c=Pluie, alpha=0.7, cmap='Blues', marker='o', transform=ccrs.PlateCarree()) # 5. Ajouter les noms des stations et les valeurs de précipitation for i, station in enumerate(station_names): plt.text(Longitude.iloc[i] + 0.1, Latitude.iloc[i] + 0.1, station, transform=ccrs.PlateCarree(), fontsize=8, ha='right', color='black', weight='bold') # Ajout de la barre de couleur plt.colorbar(scatter, ax=ax, label='Pluie Totale')

Question de Réflexion : Comment la répartition des précipitations pourrait-elle influencer les conditions climatiques et les ressources en eau dans les différentes régions du pays ?

Comparaison des Données d'Observation avec ERA5 Nous allons Télécharger des Données de précipitations journalières de ERA5 pour la même période du 1er avril au 10 octobre Pour celà, connecter vous sur le Copernicus Climate Data Store pour obtenir une clé API, puis téléchargez les données de précipitations horaires pour la même période (1er avril au 10 octobre).

Travail à faire

Comparer les Données de Station et de Réanalyse:superposez les données ERA5 aux données de station pour visualiser les différences.

Question de Réflexion : Quelle pourrait être l'impact des écarts entre ces deux types de données sur la prise de décision dans les politiques de gestion des ressources en eau ?

Exploration Supplémentaire Calcul des erreurs moyennes : Utilisez des indicateurs comme le biais ou l'erreur quadratique moyenne pour quantifier les écarts.

Études climatiques plus approfondies : Étudiez les variations saisonnières des précipitations en fonction de la localisation géographique et des périodes. Extension à d'autres variables climatiques : Appliquez les techniques utilisées ici pour analyser la température, l'humidité ou le vent.

Exploration Supplémentaire Pour approfondir votre analyse, voici quelques pistes supplémentaires :

Calcul des erreurs moyennes : Utilisez des indicateurs comme le biais ou l'erreur quadratique moyenne (RMSE) pour quantifier les écarts. Ce type de mesure vous donnera une idée de la précision de la réanalyse pour chaque station.

from sklearn.metrics import mean_squared_error import numpy as np rmse = np.sqrt(mean_squared_error(data_station['precipitation'], data_era5['precipitation'])) print(f"RMSE entre les données de station et ERA5: {rmse} mm")

Études climatiques plus approfondies : Examinez les variations saisonnières en calculant les moyennes de précipitations pour différentes saisons ou mois. Cela peut révéler des écarts spécifiques à certaines périodes.

Étendre l'analyse à d'autres variables climatiques : les données ERA5 contiennent plusieurs variables utiles pour une analyse climatique complète.

Conseil : Pour visualiser les données géospatiales, cartopy vous permet de créer des cartes montrant les cumuls de précipitations pour différentes périodes. Exemple d'Extension : Visualiser les Précipitations par Station import cartopy.crs as ccrs import cartopy.feature as cfeature # Exemple de visualisation avec cartopy fig, ax = plt.subplots(figsize=(10, 8), subplot_kw={'projection': ccrs.PlateCarree()}) ax.set_extent([-5, 5, 9, 16]) # Définir l'étendue de la carte # Ajouter les frontières et la côte ax.add_feature(cfeature.BORDERS, linestyle='-', alpha=0.5) ax.add_feature(cfeature.COASTLINE, linestyle=':', alpha=0.5) # Afficher les données de précipitations sur la carte scatter = ax.scatter(data_station['longitude'], data_station['latitude'], s=data_station['precipitation']*10, c=data_station['precipitation'], cmap='Blues', alpha=0.6, transform=ccrs.PlateCarree()) plt.colorbar(scatter, label='Précipitation (mm)') plt.title("Précipitations par Station") plt.show()

Pour comparer les données d'observation avec les données de réanalyse ERA5, nous allons télécharger les précipitations journalières d'ERA5 pour la période du 1er avril au 10 octobre. Pour ce faire, nous utiliserons le Copernicus Climate Data Store (CDS), qui nécessite une clé API pour l'accès aux données. Une fois cette clé obtenue, nous utiliserons cdsapi pour télécharger les données directement depuis Python, et xarray pour traiter les données NetCDF.

Étapes pour Télécharger et Analyser les Données ERA5 Créer un Compte sur le Portail CDS Accédez au portail Copernicus Climate Data Store, créez un compte si vous n'en avez pas déjà un, et obtenez une clé API pour accéder aux données.

Installer la Bibliothèque cdsapi et Configurer l'API Installez cdsapi en utilisant la commande suivante dans votre terminal ou environnement Jupyter :

bash Copier le code pip install cdsapi Ensuite, configurez votre clé API en ajoutant un fichier .cdsapirc dans votre répertoire personnel, contenant vos identifiants CDS sous la forme :

makefile Copier le code url: https://cds.climate.copernicus.eu/api/v2 key: votre_identifiant:clé_api Télécharger les Données de Précipitations avec cdsapi Utilisez le code suivant pour télécharger les précipitations horaires pour la période du 1er avril au 10 octobre en spécifiant la variable total_precipitation. Ce fichier sera téléchargé au format NetCDF.

python Copier le code import cdsapi

c = cdsapi.Client()

c.retrieve('reanalysis-era5-single-levels', { 'product_type': 'reanalysis', 'variable': 'total_precipitation', 'year': '2023', # Remplacez par l'année appropriée 'month': ['04', '05', '06', '07', '08', '09', '10'], 'day': [str(i).zfill(2) for i in range(1, 32)], # Inclut tous les jours 'time': ['00:00', '06:00', '12:00', '18:00'], # Précipitations horaires 'area': [16, -5, 9, 5], # Coordonnées du Burkina Faso [Nord, Ouest, Sud, Est] 'format': 'netcdf', }, 'era5_precipitation_data.nc') Chargement et Traitement des Données avec xarray Une fois le téléchargement terminé, nous utiliserons xarray pour charger le fichier NetCDF et manipuler les données.

python Copier le code import xarray as xr

Charger les données de précipitations ERA5

era5_data = xr.open_dataset('era5_precipitation_data.nc')

Extraire les précipitations journalières en les cumulant

era5_daily_precip = era5_data['total_precipitation'].resample(time='1D').sum()

Filtrer pour la période d'intérêt

era5_precip_period = era5_daily_precip.sel(time=slice("2023-04-01", "2023-10-10"))

print(era5_precip_period) Visualisation et Analyse Avec xarray, nous pouvons maintenant analyser et comparer les précipitations journalières d'ERA5 à nos données de station en utilisant des statistiques et des visualisations.

Note : xarray permet de manipuler et de visualiser facilement les données multidimensionnelles, telles que les séries temporelles des précipitations. Nous utiliserons cette bibliothèque pour extraire des statistiques, des moyennes, et pour faciliter la comparaison avec les données d'observation.