Un Peuple-Un But-Une Fois

MINISTERE DE L'ECONOMIE DU PLAN ET DE LA COOPERATION

****

Agence nationale de la Statistique et de la Démographie (ANSD-

Ecole nationale de la Statistique et de l'Analyse économique Pierre

Résumé des travaux pratiques de statistique exploratoire spatiale

Rédigé par :

Mme Edima BIYENDA HILDEGARDE

M. Papa Amadou NIANG

M. Mame Balla BOUSSO

M. Ameth FAYE

Sous la supervision de :

Aboubacar HEMA

Research analyst à IFPRI

Année Académique :

2024-2025

Table des matières

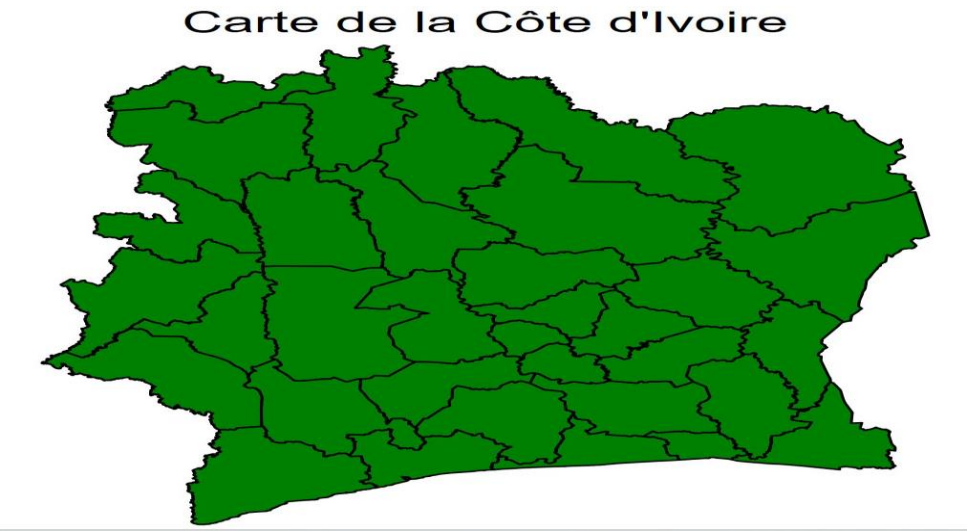
I. Travail pratique 1 3
II. TP2 (GEE-PYTHON) 4
III. TP3 et TP4 (R) 6
IV. TP5 (GEE) 8
V. TP6 - Statistique Exploratoire Spatiale (GEE avec JavaScript) 11
VI. TP6 (PYTHON) 13

I. Travail pratique 1

Dans ce TP, nous avons travaillé avec les données de la Côte d'Ivoire.

Nous avons débuté ce TP avec Stata. Les packages nécessaires étaient **shp2dta** et **spmap**. *shp2dta* nous a permis de convertir le shapefile en format Stata, tandis que le package *spmap* a permis la création de la carte.

Défis rencontrés : Stata n'est pas encore un logiciel performant en visualisation de données spatiales. Malgré cela, nous avons quand même essayé de faire la visualisation suivante.



La suite de ce TP a été réalisée avec Google Earth Engine (GEE) en utilisant Python.

Les principaux résultats obtenus sont :

- Le système de coordonnées (CRS) utilisé est **EPSG:4326**.
- La taille des pixels définie est de **1 km**.

Défis rencontrés : Le langage Python, contrairement au langage JavaScript, n'est pas directement compatible avec l'environnement de GEE. Il a donc été nécessaire d'établir une liaison entre l'environnement GEE et l'environnement Jupyter afin de pouvoir travailler avec des codes Python.

II. TP2 (GEE-PYTHON)

Dans ce TP, nous avons travaillé avec les données du Sénégal.

Nous avons réalisé une présentation détaillée pour illustrer la démarche nécessaire afin d'établir une liaison entre l'environnement Google Earth Engine (GEE) et l'environnement Jupyter. Cette configuration permet d'exploiter les fonctionnalités de GEE tout en utilisant le langage Python.

```
conda create -n gee python=3.8

conda activate gee

pip install earthengine-api geemap windows-curses

conda install jupyter

conda install ipykernel

python -m ipykernel install --user --name gee --display-name "Python (gee)"

jupyter notebook
```

Cette connexion entre ces deux environnements susmentionnés est finalisée par l'authentification suivante :

```
# Authentifier et initialiser Google Earth Engine
ee.Authenticate()
ee.Initialize()
```

Les travaux pratiques n°2 sont constitués de deux sections. La première section était consacrée à la manipulation de données vectorielles.

SECTION 1 : MANIPULATION DES DONNÉES VECTORIELLES

La première section des travaux pratiques est dédiée à la manipulation des données vectorielles en utilisant Google Earth Engine (GEE). Les principales étapes et fonctions utilisées sont les suivantes :

Importation des données vectorielles : Les données administratives du Sénégal ont été importées sous forme de collections d'entités géographiques (FeatureCollection), représentant les pays, régions, départements et communes.

Calculs de statistiques :

- **Nombre de géométries** : La fonction size() a été utilisée pour compter le nombre d'entités dans une collection.
- **CRS (système de référence de coordonnées)** : La méthode geometry().projection() permet d'obtenir les informations sur le système de coordonnées des entités géographiques.
- **Étendue (extent)** : La méthode geometry().bounds() calcule les limites géographiques des entités, fournissant ainsi l'étendue de chaque entité.
- **Centroïdes** : La fonction centroid() génère les centroïdes des géométries, utiles pour localiser les entités géographiques sur la carte.
- **Aire et périmètre** : Les fonctions area() et perimeter() ont été utilisées pour calculer respectivement l'aire et le périmètre des entités géographiques.

Visualisation : Une carte interactive a été créée pour visualiser les entités géographiques et leurs centroïdes. La méthode addLayer() permet d'ajouter des couches à la carte, et la méthode centerObject() est utilisée pour centrer la vue sur les entités importées, facilitant ainsi l'exploration des données géospatiales.

SECTION 2 : MANIPULATION DES DONNÉES RASTER

La deuxième section est consacrée à la manipulation des données raster, particulièrement les indices de parasitisme liés au paludisme au Sénégal pour la période 2000-2022. Les étapes principales sont les suivantes :

Chargement des données raster :

- Une fonction personnalisée, get_raster_by_year, permet d'importer les rasters annuels en sélectionnant la première bande de chaque image.
- Une collection d'images (ImageCollection) est ensuite créée à partir des rasters chargés.

Calculs statistiques sur la collection :

- **Moyenne** : La méthode mean() calcule une image moyenne à partir de la collection.
- **Médiane** : La fonction median() est utilisée pour extraire l'image médiane de la collection.
- **Écart-type** : La fonction reduce(ee.Reducer.stdDev()) permet de calculer l'écart-type pixel par pixel.
- **Minimum et maximum** : Les fonctions min() et max() permettent d'obtenir respectivement les valeurs minimales et maximales des pixels dans la collection.

Visualisation :

- Chaque statistique calculée est visualisée en tant que couche sur une carte interactive, avec des paramètres spécifiques de couleur (vis_params).
- Une barre de couleurs a été ajoutée pour mieux interpréter les indices calculés.

Défis rencontrés : Dans un premier temps, nous avons remarqué des valeurs manquantes dans nos données. Après inspection, nous avons constaté que la deuxième bande des rasters contenait ces valeurs manquantes. Nous avons alors décidé de sélectionner uniquement la première bande des rasters, ce qui a résolu le problème.

III. TP3 et TP4 (R)

Dans ces travaux pratiques, nous avons travaillé avec les données du Sénégal.

Le **TP3** est consacré à une analyse des données spatiales en utilisant R pour calculer des moyennes pondérées par région, département et commune à partir des rasters représentant les taux de parasitisme du paludisme au Sénégal. Les étapes principales sont les suivantes :

- **Chargement des bibliothèques et des données** :
    - Importation des rasters et shapefiles des régions, départements et communes.
    - Création d'un stack de rasters à partir des fichiers importés.
- **Calculs statistiques** :
    - Moyenne pondérée des indices de paludisme par région, département et commune en utilisant la fonction exact_extract().
- **Visualisation** :
    - Création de cartes thématiques pour chaque niveau administratif (région, département, commune) en utilisant ggplot2.
    - Visualisation interactive avec leaflet, permettant une exploration dynamique des données.
    - Ajout d'annotations comme l'échelle et la flèche du nord pour améliorer l'interprétation des cartes.

Défis rencontrés : Comme le professeur avait attiré notre attention sur les rasters pouvant être à l'intersection de deux zones pour un même niveau administratif, nous avons eu le défi de trouver une fonction permettant de faire les calculs en pondérant selon l'aire pour contourner ce problème. Nous avons trouvé que la fonction exact_extract() faisait exactement ce travail. Pour nous rassurer, nous avons consulté la documentation du package exactextractr, particulièrement la fonction exact_extract.

Le **TP4**, quant à lui, a abordé la classification des rasters de paludisme des années 2020, 2021 et 2022 en trois catégories : "Aucun", "Moyen" et "Grave". Cette classification visait à mieux comprendre les zones présentant différents niveaux de gravité de la maladie. Des seuils basés sur la moyenne et l'écart-type des valeurs des rasters ont été définis pour créer ces catégories. Une

fonction a été développée pour classifier chaque pixel du raster selon ces seuils, et des cartes interactives ont été générées avec les bibliothèques raster et leaflet. Les résultats ont permis de créer des cartes colorées représentant les zones à faible, moyen et haut risque de malaria, facilitant ainsi une gestion ciblée des risques.

En conclusion, ces deux travaux pratiques ont permis d'utiliser les données géospatiales pour analyser la répartition du paludisme au Sénégal. Les cartes générées offrent une visualisation claire des zones les plus à risque, ce qui est essentiel pour la planification et l'orientation des politiques sanitaires dans le pays. Ces analyses permettent une gestion plus efficace des interventions contre le paludisme, en ciblant spécifiquement les zones les plus affectées.

IV. TP5 (GEE)

Dans le TP5, il était question d'importer et de visualiser les données de WorlPop, puis de calculer le nombre de personne par les divisions administratives (Pays, Région, département, commune). Le raster Worlpop utilisé dans ce TP contenait la population par pixel de 100m.

Dans les TP précédents, nous avions relevé les endroits où, il y avait une prévalence ou non du taux de Malaria des enfants de moins de 12ans, selon un critère de classification, dépendant de la moyenne et de l'écart type. A cet effet trois endroits labélisés aucune situation grave, situation modérée et situation grave ont été obtenus. Ces statistiques étaient, comme nous l'avons dit, des taux. Mais pour connaître le nombre d'enfants exact atteint de Malaria dans chaque situation, il fallait utiliser le raster Worlpop qui contient le nombre de personne par raster et avec l'hypothèse que les enfants de moins de 12 ans étaient estimés à 0,1% de la population dans chaque pixel, il était simpliste de connaître le nombre d'enfants, et avec les taux, le nombre d'enfants malades. Ce dernier a été déterminé pour chaque situation. Ce raster a aussi permis de connaitre le nombre d'enfants par division administrative, avec une sortie CSV, donnant un featureCollection. La résolution du raster qui était préalablement de 100m par pixel, nécessitait par ailleurs d'être ramené à 5km. Ce qui fait que dans chaque raster de 5km, nous avons le nombre de personne conformément à ceux contenant le taux de Malaria. A la fin, disposant des feature Collection contenant d'une part,

le nombre de personne par division administrative et d'autre part le nombre d'enfants atteint de Malaria par division administrative, le taux d'enfants atteint de Malaria a été aussi calculé.

Il est important de noter qu'ici, dans l'impossibilité de manipuler les deux featuer collection par la division des deux colonnes concernées, nous étions obligés de créer, dans un feature collection (celui contenant le nombre total de personne par exemple), une autre bande contenant les le nombre d'enfants malades. Le feature collection obtenu obtenu a donc été composé de deux bandes et que la division a été faite directement, permettant d'avoir le résultat escompté.

Avec google earth engine, *reduceRegions* a permis de déterminer les populations par division administrative en sommant avec *ee.Reducer.sum()*.

Pour changer la résolution, nous avons fait un *reproject* avec : *reproject({ crs: population.projection(), scale: 5000 })* précisant le crs et la résolution visée qui est ici de 5km.

Le raster obtenu a été utilisé comme ceci *population_5km.multiply(0.001)* pour obtenir la population de moins de 12 ans par pixel.

Pour déterminer les trois situations décrites au-dessus, il fallait d'abord un critère de classification, mais aussi une situation de référence. A ce sens, nous avons défini les seuil 1,2 et 3 avec la moyenne et l'écart type avec la situation de 2022 comme référence (ras_cible).

On a ainsi créé les trois situations avec raster1 correspondant aux situations où le taux de Malaria en 2022 dépasse le seuil1 (qui est la somme entre moyenne et écart type des taux de 2000 à 2022), raster2 donnant les situations qui chevauchent entre les seuil1 et seuil2 (qui donne la moyenne plus deux fois l'écart type de 2000 à 2022) et raster2 présentant les taux qui dépassent le seuil2.

```
var seuil1 = ras_moy.add(ras_ect);

var seuil2 = ras_moy.add(ras_ect.multiply(2));

var aucun = ras_cible.lt(seuil1);

var modere = ras_cible.gt(seuil1).and(ras_cible.lt(seuil2));

var grave = ras_cible.gt(seuil2);
```

Les rasters *aucun*, *modere* et *grave* étant binaires (0 et 1) leur multiplication avec le raster *population_0_1* qui contient les populations de moins de 12 ans, donnent directement la population d'enfants de moins de 12 ans dans chaque cas.

```
var raster1 = aucun.multiply(population_0_1);
var raster2 = modere.multiply(population_0_1);
var raster3 = grave.multiply(population_0_1);
```

En outre, on a utilisé une fonction *calcetexportpop* qui prend pour arguments un raster, un shapefile contenant la division administrative concernée et une description qui permet de nommer le feature collection obtenu. La fonction sera appliquée avec le raster *malaria_pop = ras_cible.multiply(population_0_1)* qui donne la population malade en 2022.

En fin, pour calculer le taux de Malaria dans la population totale, il fallait faire la division entre la population malade déjà obtenue par celle globale. Dans l'impossibilité, comme mentionnée plus haut, de faire la division directement en utilisant les deux feature collection, nous avons ajouté une autre bande par *malaria_pop.addBands(population_0_1)* pour ensuite faire la division en interne. On a récupéré ensuite les bandes et fait la division comme ainsi qui suit :

```
var malaria = ee.Number(feature.get('b1'));
var population = ee.Number(feature.get('b1_1'));
var taux = malaria.divide(population).multiply(100);
```

V. TP6 - Statistique Exploratoire Spatiale (GEE avec JavaScript)

Le travail pratique avait pour objectif d'explorer des données géospatiales et temporelles au Sénégal en suivant plusieurs consignes : importer les points et limites administratives, compter le nombre de points par niveau administratif (pays, régions, départements, communes), créer un raster de 5 km regroupant les points, visualiser le raster sous forme de catégories, et analyser l'évolution annuelle des événements. Les données utilisées comprenaient des limites administratives détaillées (ADM0 à ADM3) et des points représentant des événements géolocalisés associés à des années spécifiques.

Dans un premier temps, les points ont été filtrés pour ne conserver que ceux localisés au Sénégal. Un comptage a été effectué pour chaque niveau administratif, permettant de quantifier les événements par pays, région, département et commune. Ensuite, un raster à une résolution de 5 km a été créé pour agréger les points dans chaque pixel.

Enfin, des rasters annuels ont été générés pour chaque année présente dans les données, permettant une analyse temporelle fine. Le nombre total d'événements par année a été calculé et représenté dans un graphique d'évolution, mettant en évidence les variations annuelles. Ces étapes ont permis de combiner des analyses spatiales et temporelles, renforçant les compétences en visualisation et en traitement de données géospatiales.

VI. TP7 (R)

I. Installation de l'application sdmApp

1) Préparation de l'environnement Java :

- Installation de Java JDK 23, vérification de son installation, et configuration pour R.

2) Installation des dépendances :

- Packages nécessaires (rJava, CENFA, remotes, etc.) installés et configurés.

3) Installation et lancement de sdmApp :

- Package installé via GitHub avec ses dépendances et lancé avec sdmApp().

II. Présentation des différents onglets de l'application

1) À propos de sdmApp :

- Présentation générale de l'application comme un outil interactif basé sur Shiny pour modéliser la distribution des espèces.

2) Onglet Data Upload :

- Chargement des données environnementales et d'occurrence des espèces.
- Visualisation des rasters et des données d'occurrence avec personnalisation des graphiques et export possible.

3) Onglet Spatial Analysis :

- Résumé des données environnementales et d'occurrence avec options de recherche et tri.
- Analyses avancées comme les corrélations environnementales, niches écologiques (ENFA), et partitionnement spatial.

4) Onglet Modeling :

- **Choix du modèle** : Sélection parmi plusieurs algorithmes, tels que MaxEnt, Random Forest, Support Vector Machines, etc.
- **Configuration** : Définir les paramètres, comme le nombre de "folds" pour la validation croisée.
- **Visualisation des résultats** : Cartes de prédictions, analyse de l'importance des variables et réponses environnementales.
- **Exportation** : Possibilité de télécharger les résultats en .png, .pdf ou .tif.

5) Onglet R-Code :

- **Affichage du script généré** : Contient les commandes R pour reproduire les analyses.
- **Téléchargement** : Enregistrer le script pour documentation ou modifications ultérieures.

VII. TP10 (PYTHON)

Le TP10 avait pour objectif de calculer des indices spectraux basés sur les bandes spectrales. Ainsi, nous avons d'abord téléchargé sur COPERNICUS des données rasters en sélectionnant les huit premières bandes. Nous avons utilisé des fichiers TIFF du satellite Sentinel-2 et visualisé ces indices pour en faciliter l'interprétation. À l'aide d'un script, nous avons parcouru le dossier pour lire chaque fichier, identifier les bandes (B01 à B08) à partir des noms de fichiers et les charger en mémoire sous forme de tableaux numpy au format flottant (float32). Nous avons également mis en place une étape de validation pour vérifier que toutes les bandes nécessaires aux calculs étaient présentes.

Ensuite, nous avons attribué chaque bande spectrale à une variable distincte (b01, b02, ..., b08) pour faciliter nos calculs. Nous avons calculé des indices spectraux couramment utilisés en télédétection. Parmi eux : le NDVI (Normalized Difference Vegetation Index) pour évaluer la

végétation saine, l'ANDWI (Automated Normalized Difference Water Index) pour détecter l'eau tout en minimisant les effets de la végétation, le BRBA (Burnt Ratio Brightness Adjusted) pour identifier les zones brûlées, le GNDVI (Green Normalized Difference Vegetation Index) pour analyser la concentration de chlorophylle, le SAVI (Soil-Adjusted Vegetation Index) qui améliore l'analyse de la végétation dans les zones où le sol nu est visible, le kIPVI (Kernel IPVI) pour mesurer la réflectance relative à la végétation, et le MLSWI26 (Modified Land Surface Water Index) pour identifier les zones humides à partir des bandes du visible et de l'infrarouge.

Pour ces calculs, nous avons appliqué des formules mathématiques pixel par pixel aux bandes spectrales. Afin d'éviter les divisions par zéro et les valeurs manquantes, nous avons utilisé une fonction pour remplacer ces cas par des zéros (np.nan_to_num). Une fois les indices calculés, nous avons utilisé une fonction personnalisée pour visualiser les résultats sous forme de cartes spatiales. Ces cartes, affichées avec des échelles de couleurs adaptées (par défaut, cmap='RdYlGn'), nous ont permis d'interpréter les variations des indices dans l'espace.

En somme, ce TP nous a permis de manipuler des données raster, d'extraire des informations pertinentes à partir des bandes spectrales Sentinel-2 et de produire des visualisations utiles pour l'analyse environnementale, notamment pour détecter la végétation, les zones d'eau et les surfaces brûlées.