UK Centre for Ecology & Hydrology

Observation de la Terre pour la modélisation du paludisme : une boîte à outils pratique pour la prédiction par satellite de la distribution des moustiques à l'aide de Google Earth Engine et de R

Date 08/02/2023

Christopher G. Marston[1], Clare S. Rowland[1], Aneurin W. O'Neil[1], Seth Irish[2], Francis Wat'senga[3], Pilar Martin-Gallego[4], Patrick Giraudoux[5], Clare Strode[4].

[1] UK Centre for Ecology & Hydrology, Library Avenue, Bailrigg, Lancaster. United Kingdom. LA14AP.

[2] U.S. President's Malaria Initiative and Centers for Disease Control and Prevention, 1600 Clifton Road NE, Atlanta, GA, 30329. USA.

[3] Institut National de Recherche Biomédicale (INRB), Avenue de la Démocratie Nº 5345, Kinshasa – Gombe, Democratic Republic of the Congo.

[4] Edge Hill University, St Helens Road, Ormskirk, Lancashire L394QP, United Kingdom.

[5] Department of Chrono-Environment, University of Bourgogne Franche-Comte/CNRS, La Bouloie, 25030 Besançon CEDEX, France.

Edge Hill University

CDC Centers for Disease Control and Prevention

INRB Institut National de Recherche Biomédicale Laboratoire National de Santé Publique

Documentation du jeu de données

Version 1.1, 08/02/2023

Version	Date	Mises à jour
1.1	08/02/2023	Version originale en langue française

Contenu

1 Introduction

Ce guide d'utilisation accompagne la publication de Marston *et al.* (2023). Il fournit une introduction conviviale à la mise en œuvre de l'analyse par satellite et de la modélisation par *Random Forest* pour **identifier les variables bio-géographiques clés qui influencent la distribution et l'abondance des moustiques**. Il est destiné à servir de ressource pour les utilisateurs ayant des connaissances préalables limitées de ce type d'analyse et présente des instructions étape par étape permettant aux utilisateurs d'effectuer une modélisation prédictive des distributions de moustiques ; des exemples de jeux de données et de scripts d'analyse sont fournis.

Ce guide utilise trois logiciels, Google Earth Engine, R (avec RStudio) et QGIS pour le prétraitement, la modélisation et la visualisation des données, qui sont tous gratuits pour un usage non commercial. Des scripts sont fournis pour effectuer le traitement et l'analyse des données à la fois dans Google Earth Engine (GEE) et dans R. Bien que ces scripts soient conçus pour automatiser l'analyse dans une large mesure, ils sont actuellement optimisés pour la zone d'étude et la période de temps utilisées dans l'exemple présenté (c'est-à-dire Lodja, République démocratique du Congo). L'utilisateur devra adapter les scripts à des zones d'étude et périodes d'intérêt différentes, ce qui nécessitera alors de disposer des données d'enquête correspondantes sur les moustiques avec les coordonnées de localisation associées (c'est-à-dire les degrés décimaux de latitude et de longitude) et les dates. Les utilisateurs sont libres d'adapter les exemples présentés sur cette base pour répondre à leurs propres besoins.

Ce manuel d'utilisation comprend les sections suivantes :

- Introduction
- Installation et enregistrement du logiciel
- Zone d'étude
- Génération de jeux de données - QGIS
- Google Earth Engine - prétraitement des données satellite
- R - sélection des caractéristiques
- Google Earth Engine - modélisation
- Google Earth Engine - visualisation des données

Ce manuel de l'utilisateur est fourni avec les fichiers suivants :

- Script R
- Données de l'enquête de terrain sur les moustiques de Lodja, RDC (fournies dans le script GEE).
- Trois scripts Google Earth Engine (GEE)
- Données de formation sur la couverture du sol (fournies comme ressource dans le script GEE).

En suivant les instructions, l'utilisateur apprendra à appliquer les méthodes requises, mais produira également des fichiers clés nécessaires aux étapes suivantes. Ces fichiers peuvent être utiles si l'utilisateur souhaite appliquer ces méthodes à desensembles de données, zones et scénarios différents de son choix. Le flux de travail du traitement des données implique de passer d'un logiciel à l'autre à différents stades de l'analyse. La Figure 1.1 présente une vue d'ensemble des principales étapes du traitement et des logiciels correspondants utilisés

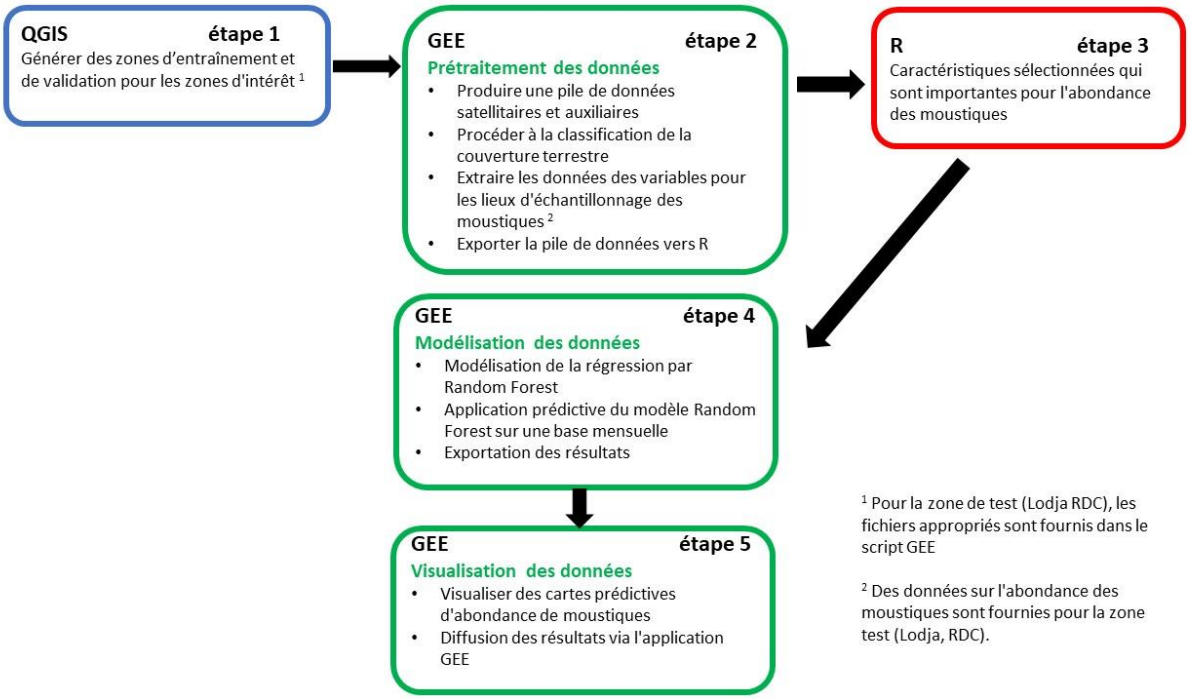


Figure 1.1. Vue d'ensemble des principales étapes du traitement des données et des logiciels correspondants utilisés.

2 Installation et enregistrement du logiciel

Pour mettre en œuvre la fonctionnalité de ce manuel de l'utilisateur, il est nécessaire d'installer QGIS et RStudio sur votre ordinateur et de créer un compte utilisateur pour Google Earth Engine pour le traitement des données dans le cloud. La documentation suivante est destinée à un système d'exploitation Windows 64 bits. Les instructions d'installation pour d'autres systèmes d'exploitation peuvent différer - dans ce cas, veuillez-vous reporter aux instructions d'installation des progiciels respectifs pour votre système d'exploitation. Le traitement et l'analyse contenus dans ce guide ont nécessité une quantité importante d'espace de stockage. Nous vous recommandons de vous assurer que vous disposez d'au moins 10 Go d'espace libre avant d'effectuer l'analyse.

2.1 Installation de QGIS

Allez sur le site web de QGIS - https://www.qgis.org/en/site/ (Figure 2.1) et cliquez sur le bouton "Download Now" (Télécharger maintenant)

Figure 2.1. Site Web de QGIS

Dans la fenêtre de téléchargement qui s'ouvre (Figure 2.2), sélectionnez l'option de téléchargement appropriée pour la plate-forme que vous utilisez. Ici, nous sélectionnerons la version la plus récente de QGIS standalone Installer Version 3.20

(64 bit). Les mises à jour de QGIS sont publiées à intervalles réguliers, la version la plus récente disponible peut donc différer de la version 3.20 utilisée ici. Cliquez sur le lien 'QGIS Standalone Installer Version 3.20'.

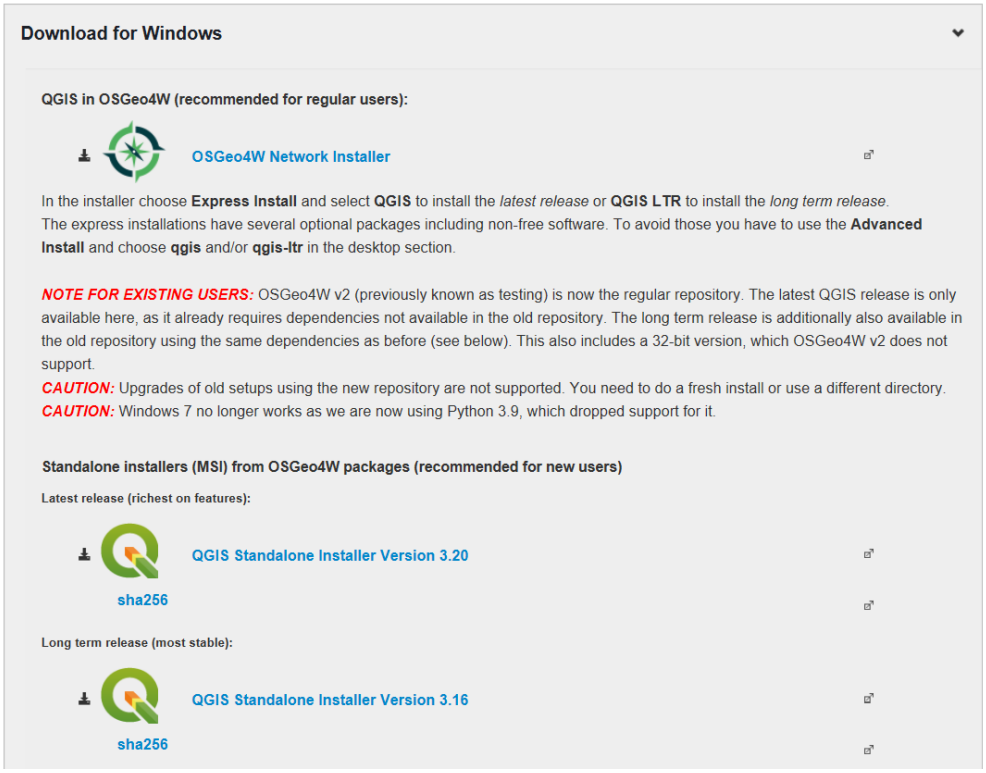


Figure 2.2. Options de téléchargement de QGIS.

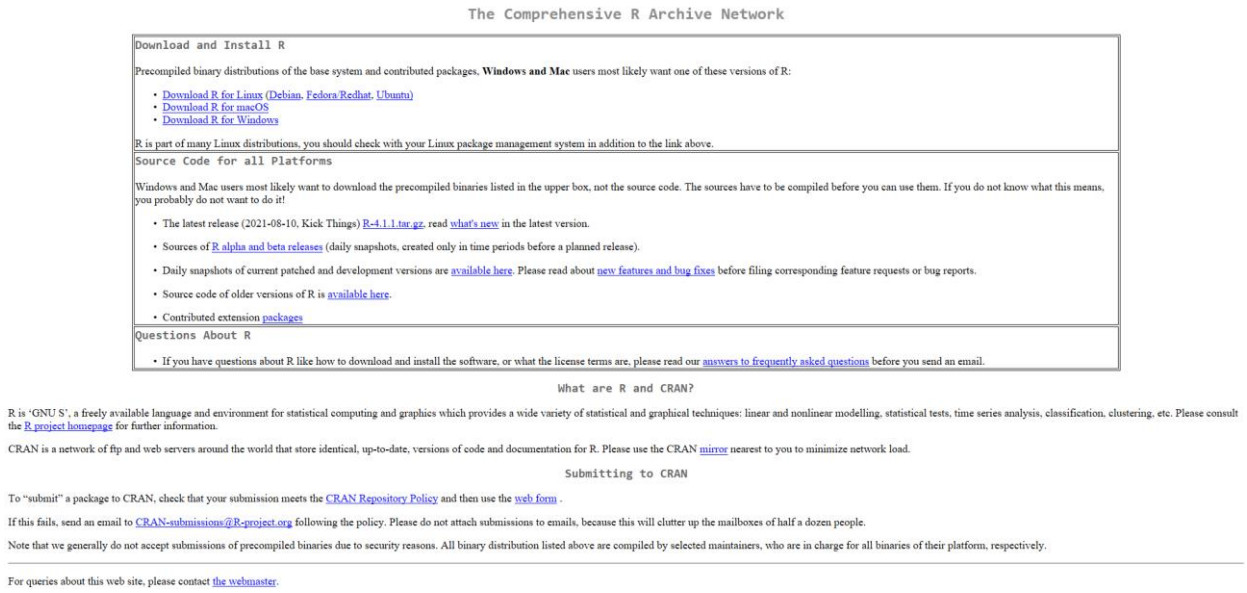
Sélectionnez "Enregistrer sous" dans le menu déroulant "Enregistrer" et sélectionnez un répertoire cible sur l'ordinateur à partir duquel l'application QGIS sera exécutée. Selon le système d'exploitation que vous utilisez, le téléchargement peut démarrer automatiquement.

Naviguez jusqu'au dossier dans lequel le fichier d'installation a été enregistré et double-cliquez dessus pour lancer le processus d'installation. Si le téléchargement a démarré automatiquement, le fichier devrait être enregistré dans le dossier "Téléchargements". Vous pouvez également cliquer sur "Exécuter" en bas du navigateur Web. L'assistant d'installation de QGIS démarre alors et vous guide tout au long du processus d'installation. Il y a une option pour télécharger trois jeux de données d'échantillon (jeu de données de Caroline du Nord, Dakota du Sud (Spearfish) et Alaska). Vous n'avez pas besoin de les télécharger pour exécuter les méthodes présentées dans ce guide de l'utilisateur, mais ils peuvent être utiles comme jeux de données de test pour explorer les fonctionnalités plus larges de QGIS.

Vous trouverez d'autres documents utiles à l'adresse suivante : https://www.qgis.org/fr/site/.

2.2 Installation de R

Allez sur le site Web de R https://cran.rstudio.com/, et sélectionnez le lien "Télécharger R pour Windows" (Figure 2.3).



The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux (Debian, Fedora/Redhat, Ubuntu)
- Download R for macOS
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2021-08-10, Kick Things) R-4.1.1.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

What are R and CRAN?

R is 'GNU S', a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the R project homepage for further information.

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN mirror nearest to you to minimize network load.

Submitting to CRAN

To "submit" a package to CRAN, check that your submission meets the CRAN Repository Policy and then use the web form .

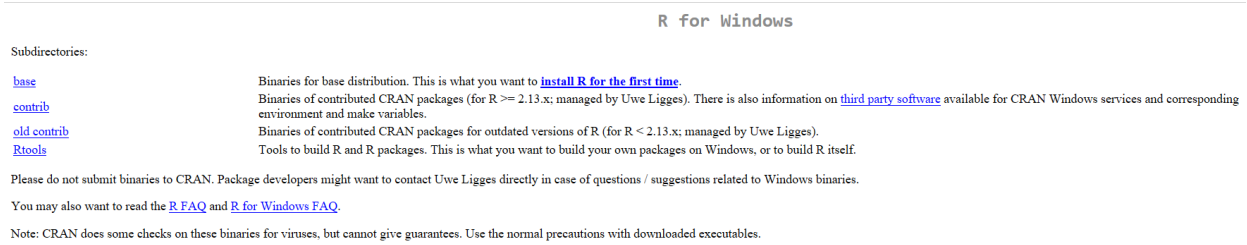
If this fails, send an email to CRAN-submissions@R-project.org following the policy. Please do not attach submissions to emails, because this will clutter up the mailboxes of half a dozen people.

Note that we generally do not accept submissions of precompiled binaries due to security reasons. All binary distribution listed above are compiled by selected maintainers, who are in charge for all binaries of their platform, respectively.

For queries about this web site, please contact the webmaster.

Figure 2.3. Site Web de Cran Rstudio avec options de téléchargement.

Sélectionnez le sous-répertoire "base" pour installer R (Figure 2.4).



R for Windows

Subdirectories:

base	Binaries for base distribution. This is what you want to **install R for the first time**.
contrib	Binaries of contributed CRAN packages (for R >= 2.13.x; managed by Uwe Ligges). There is also information on third party software available for CRAN Windows services and corresponding environment and make variables.
old contrib	Binaries of contributed CRAN packages for outdated versions of R (for R < 2.13.x; managed by Uwe Ligges).
Rtools	Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the R FAQ and R for Windows FAQ.

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

Figure 2.4. Sous-répertoires de la base de téléchargement R.

Sélectionnez le lien "Télécharger R 4.1.1 pour Windows" (ou la version la plus récente disponible si la version 4.1.1 a été actualisée) (Figure 2.5).

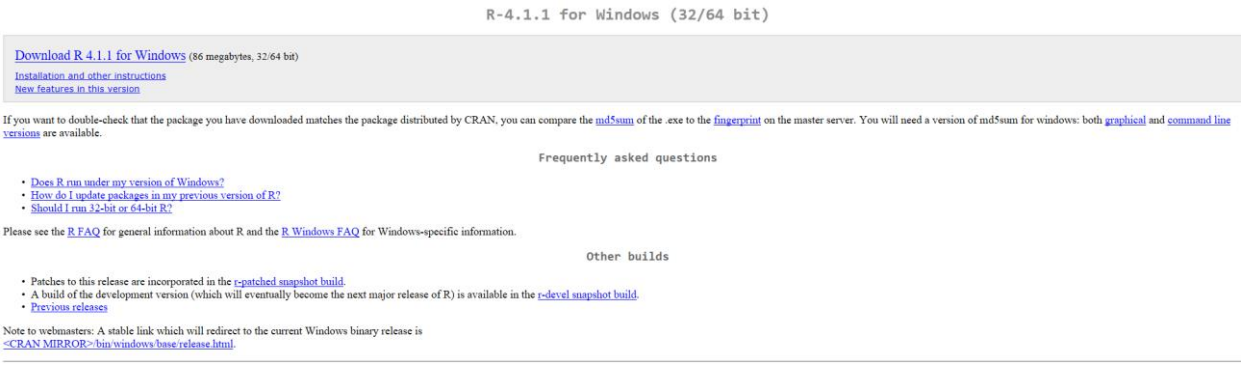


Figure 2.5. Lien de téléchargement R.

Une fois le fichier d'installation téléchargé, double-cliquez sur le fichier pour lancer le processus d'installation. L'assistant d'installation vous guidera ensuite à travers les étapes de l'installation.

2.3 Installation de RStudio

RStudio fonctionne comme front-end de R, offrant les fonctionnalités de R à travers une interface plus conviviale. RStudio doit être installé séparément, après avoir installé R.

Installation

Allez sur le site "Download RStudio" - https://www.rstudio.com/products/rstudio/download/.

Un certain nombre d'options de téléchargement sont disponibles, mais c'est le RStudio Desktop (Licence Open Source) qui est requis ici. Cette option peut être téléchargée gratuitement. Cliquez sur l'icône "Télécharger" sous cette option (Figure 2.6).

Figure 2.6. Page web de téléchargement de RStudio.

Dans la fenêtre suivante, cliquez sur l'icône "Télécharger RStudio pour Windows" (Figure 2.7).

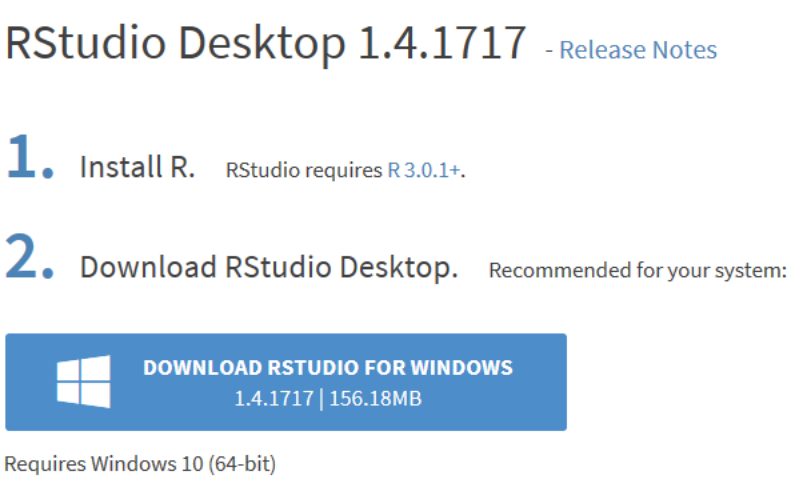


Figure 2.7. Télécharger le lien RStudio de Windows.

Enregistrez le fichier d'installation à un emplacement approprié. Une fois téléchargé, naviguez jusqu'à l'emplacement où il a été enregistré et double-cliquez sur le fichier d'installation pour commencer l'installation. Alternativement, selon le navigateur Web utilisé, une option peut apparaître pour lancer l'installation au bas de la page du navigateur Web. L'assistant d'installation de RStudio vous guidera alors à travers les étapes de l'installation.

Vous trouverez d'autres documents utiles sur les sites suivants : www.rstudio.com et https://cran.r-project.org/bin/windows/Rtools/.

2.4 Inscription au moteur Google Earth

Un enregistrement de l'utilisateur est nécessaire pour utiliser Google Earth Engine, ce qui est gratuit pour les applications non commerciales. Si vous ne disposez pas déjà d'un compte Google, il est nécessaire de s'inscrire préalablement pour obtenir un compte Google Drive.

2.4.1 Inscription à Google Drive

1. Ouvrez un navigateur Web et accédez à : accounts.google.com/SignUpWithoutGmail

2. Saisissez votre nom, votre adresse électronique et définissez un mot de passe (Figure 2.8).

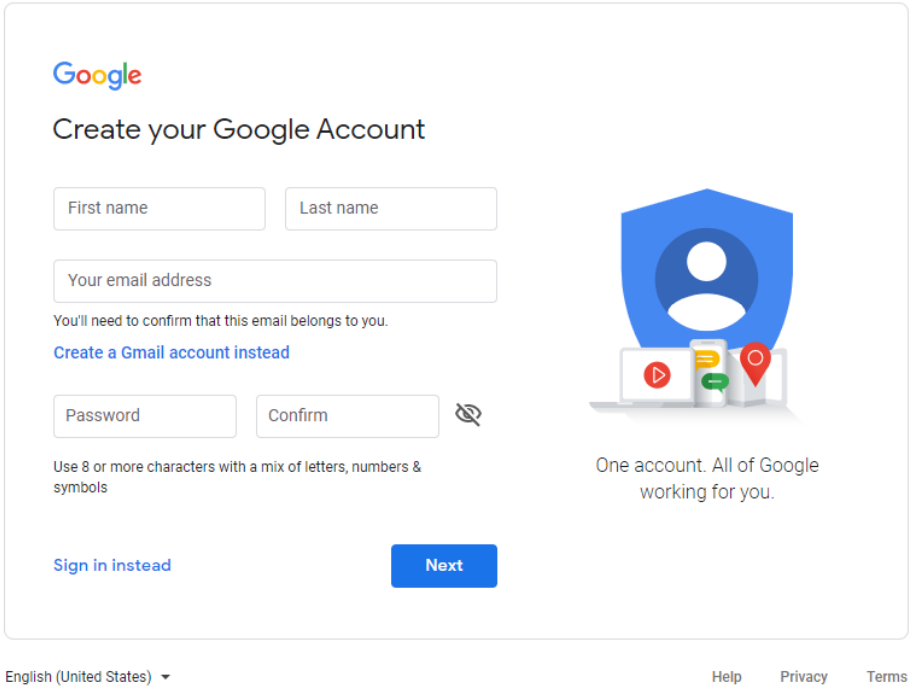


Figure 2.8. Page web d'enregistrement du compte Google.

3. Saisissez le code de vérification envoyé au compte de messagerie.

4. L'ajout d'un numéro de téléphone mobile est requis à ce stade comme autre méthode de vérification. Lorsque vous recevez un message texte contenant le code de vérification, entrez-le dans la deuxième case.

5. Remplissez la case des informations personnelles pour terminer la création du compte. Ensuite, acceptez les conditions générales, ainsi que la politique de confidentialité pour terminer la création du compte.

Pour accéder à votre compte Google Drive, saisissez l'URL suivante dans le navigateur Web - https://www.google.co.uk/drive/ et sélectionnez l'option "Go to Google Drive". Connectez-vous à l'aide de vos données de connexion nouvellement créées. Vous aurez alors accès à votre Google Drive et à son contenu. C'est là que seront exportés les jeux de données que vous créerez ensuite dans Google Earth Engine.

2.4.2 Inscription au moteur Google Earth

1. Si vous n'êtes pas encore connecté, connectez-vous à Google avec votre compte Google, puis saisissez l'URL suivante dans la barre d'URL https://earthengine.google.com/.

2. Cliquez sur le bouton "S'inscrire" en haut à droite de la page (Figure 2.9).

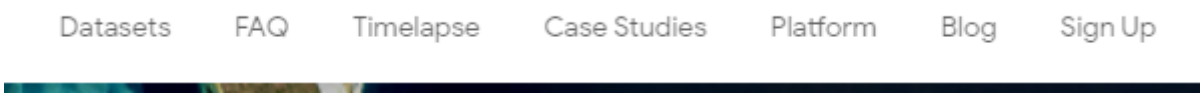


Figure 2.9. Bouton "Inscription" du moteur Google Earth.

3. Remplissez le formulaire illustré à la Figure 2.10 avec vos informations pertinentes.

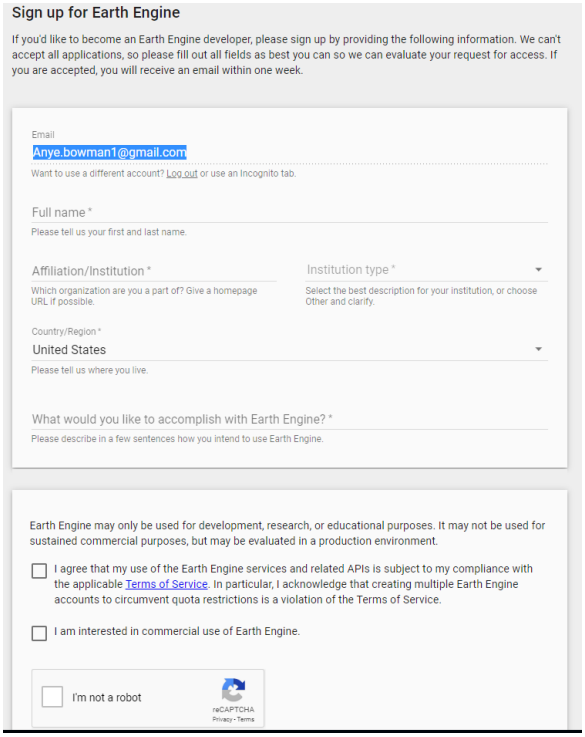
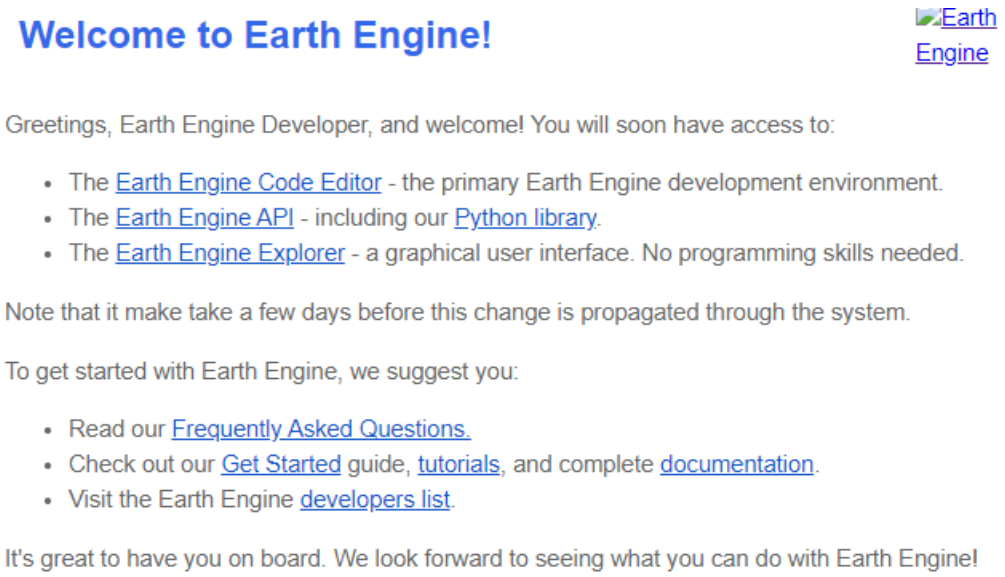


Figure 2.10. Formulaire d'inscription au moteur Google Earth.

4. Un e-mail de confirmation d'enregistrement sera envoyé au compte e-mail utilisé pour enregistrer le compte. Ouvrez l'e-mail d'enregistrement et cliquez sur le lien

"Earth Engine Code Editor" qu'il contient pour accéder à l'éditeur de code. Cet e-mail d'enregistrement peut prendre un certain temps, nous encourageons donc les utilisateurs à prévoir suffisamment de temps entre la soumission de l'enregistrement et le moment où vous devez effectuer l'analyse.

L'e-mail d'inscription (Figure 2.11) contient des liens vers l'éditeur de code de Earth Engine dans lequel l'analyse sera effectuée, mais aussi vers l'API de Earth Engine qui contient une foule d'informations de référence sur les fonctionnalités et les jeux de données de Google Earth Engine. Il contient également une variété d'autres liens, notamment vers des questions fréquemment posées et des didacticiels et de la documentation supplémentaires. Il est intéressant pour les utilisateurs de prendre le temps d'explorer ces ressources pour une introduction plus large aux capacités de Google Earth Engine.



Welcome to Earth Engine!

Earth Engine

Greetings, Earth Engine Developer, and welcome! You will soon have access to:

- The Earth Engine Code Editor - the primary Earth Engine development environment.
- The Earth Engine API - including our Python library.
- The Earth Engine Explorer - a graphical user interface. No programming skills needed.

Note that it make take a few days before this change is propagated through the system.

To get started with Earth Engine, we suggest you:

- Read our Frequently Asked Questions.
- Check out our Get Started guide, tutorials, and complete documentation.
- Visit the Earth Engine developers list.

It's great to have you on board. We look forward to seeing what you can do with Earth Engine!

Figure 2.11. Courriel d'inscription au moteur Google Earth avec les liens pertinents.

3 Zone d'étude

L'exemple présenté dans ce guide de l'utilisateur se concentre sur une zone d'étude de la ville de taille moyenne de Lodja (population d'environ 80 000 habitants) (latitude : -3,524661º, longitude : 23,596669º) et de ses environs en République démocratique du Congo, qui est méso-endémique pour le paludisme (Figure 3.1). La zone entourant Lodja est caractérisée par un mélange de composantes de la couverture terrestre, comme les cultures itinérantes traditionnelles des petits exploitants (terres défrichées, champs actifs, champs en jachère) ainsi que des zones de bâti, des zones herbeuses et dénudées, et une zone d'interface perméable avec la forêt. La rivière Lukene coule immédiatement au sud de Lodja. Les précipitations mensuelles moyennes (entre 1991-2015) étaient <100 mm pour juin et juillet et 100-220 mm/mois pour les autres mois de l'année, avec des températures moyennes de 24-26 ºC toute l'année.

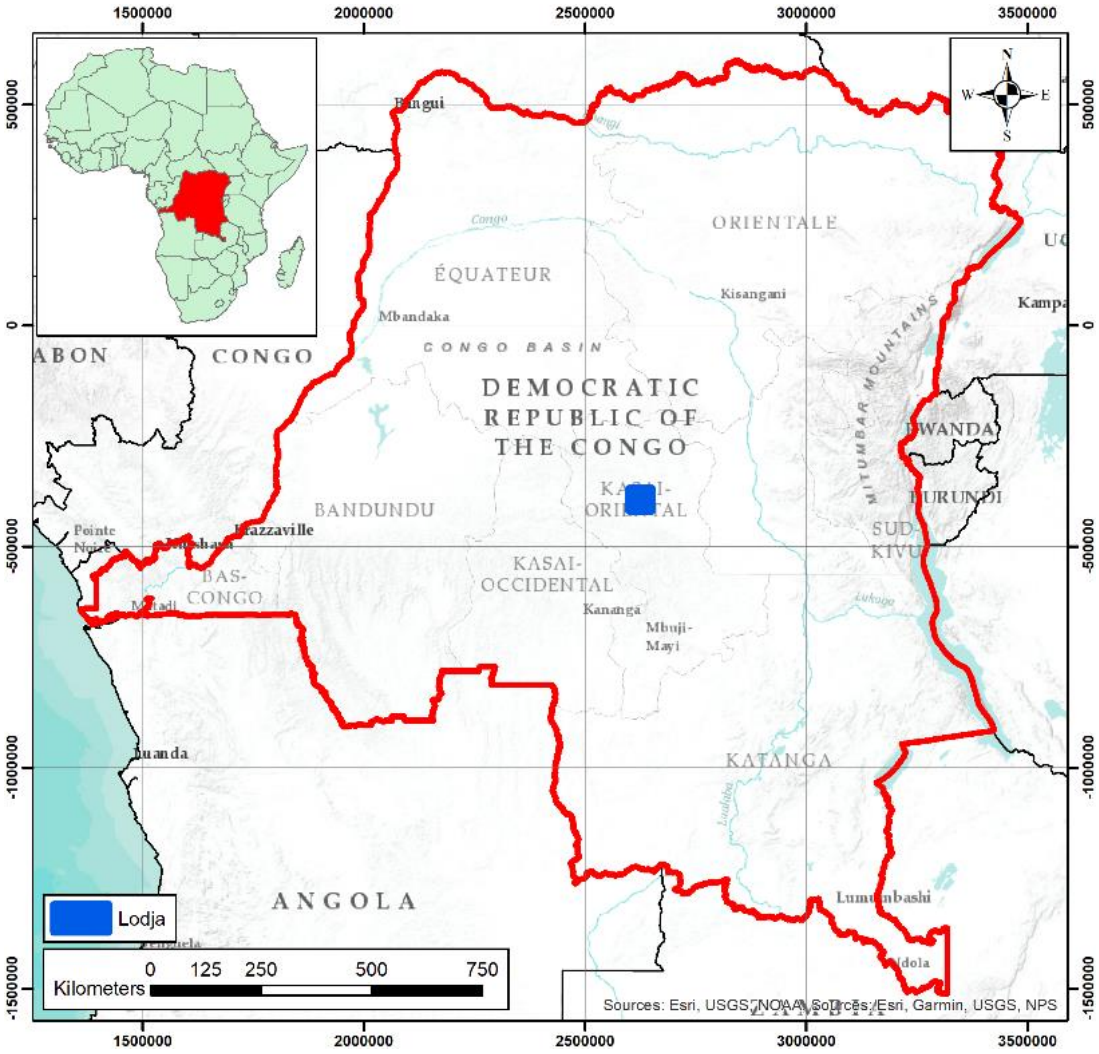


Figure 3.1 Localisation de la zone d'étude de Lodja.

4 Génération de jeux de données - QGIS

La majeure partie du flux de travail de ce manuel de l'utilisateur s'appuie sur les archives de données intégrées de Google Earth Engine, mais d'autres données d'entrée, spécifiques à la zone d'étude concernée, sont également nécessaires. Ces données comprennent

1) Des coordonnées, ou un shapefile[1] délimitant la zone d'intérêt de l'étude.

2) Un shapefile de polygones correspondant aux zones d'entrainement du modèle pour générer une classification de l'occupation du sol de la zone d'étude. Chaque polygone d'entrainement nécessite un code/attribut entier correspondant à la classe de couverture au sol correspondant à ce polygone.

3) Un ensemble de données de validation des emplacements des types de couverture terrestre connus, pour effectuer une évaluation de la précision de la classification de la couverture terrestre. Ce shapefile devrait comporter des points contenant des codes/attributs entiers qui correspondent à la classe de couverture terrestre à cet endroit. Les codes entiers utilisés pour chaque classe de couverture au sol doivent être les mêmes les shapefiles d'entrainement (polygone) et ceux de validation (point).

4) Les données de l'enquête sur les moustiques. Pour chaque emplacement : coordonnées de l'emplacement, date et espèces de moustiques et leur abondance.

Les données de comptage des moustiques doivent être fournies sous forme de fichier .csv (visualisable et modifiable dans Excel), avec des colonnes correspondant aux différents attributs de données, et des lignes correspondant aux enregistrements individuels de l'enquête (Figure 4.1).

	A	B	C	D	E	F
1	Long_dd	Lat_dd	Year	Month	House	An_gambiae_total
2	23.5830333	-3.5394833	2016	January	1	6
3	23.5830000	-3.5397167	2016	January	2	21
4	23.5836833	-3.5402833	2016	January	3	84
5	23.5837000	-3.5395333	2016	January	4	87
6	23.5838167	-3.5394000	2016	January	5	33
7	23.5839333	-3.5393000	2016	January	6	4
8	23.5833500	-3.5389167	2016	January	7	5
9	23.5999500	-3.5394167	2016	January	8	24
10	23.5830333	-3.5394833	2016	February	1	47
11	23.5830000	-3.5397167	2016	February	2	67
12	23.5836833	-3.5402833	2016	February	3	94
13	23.5837000	-3.5395333	2016	February	4	14
14	23.5838167	-3.5394000	2016	February	5	7
15	23.5839333	-3.5393000	2016	February	6	17
16	23.5833500	-3.5389167	2016	February	7	7
17	23.5999500	-3.5394167	2016	February	8	5

Figure 4.1. Exemple de format de données d'une enquête sur les moustiques.

[1] Un ***shapefile*** est un format de fichier pour les systèmes d'informations géographiques. Cf https://fr.wikipedia.org/wiki/Shapefile

4.1 Création d'un shapefile de points

Ici, nous allons créer un shapefile de points correspondant aux emplacements d'échantillonnage des moustiques sur le terrain. Les shapefiles peuvent être créés manuellement (voir la section ci-dessous où un shapefile est créé et renseigné) ou générés à partir de points de référence au sol préexistants, généralement au format feuille de calcul. Nous allons créer un shapefile de points à partir du fichier préexistant mosquito_survey_data.csv fourni avec ce tutoriel. Pour importer à partir de ce fichier .csv, suivez ces étapes :

- Ouvrez QGIS et créez un nouveau projet en cliquant sur l'icône située sous la barre d'outils principale.
- Dans la barre d'outils principale "Couche", sélectionnez "Ajouter une couche", puis "Ajouter une couche de texte délimité" (Figure 4.2). Sélectionnez le fichier .csv à importer dans la case "Nom du fichier".
- Cliquez sur l'option "Définition de la géométrie". Assurez-vous que l'option "Coordonnées du point" est sélectionnée, puis sélectionnez "Long_dd" pour le "champ X" et "Lat_dd" pour le "champ Y".
- Assurez-vous que les coordonnées X et Y sont dans les bonnes colonnes et conservez les paramètres par défaut.

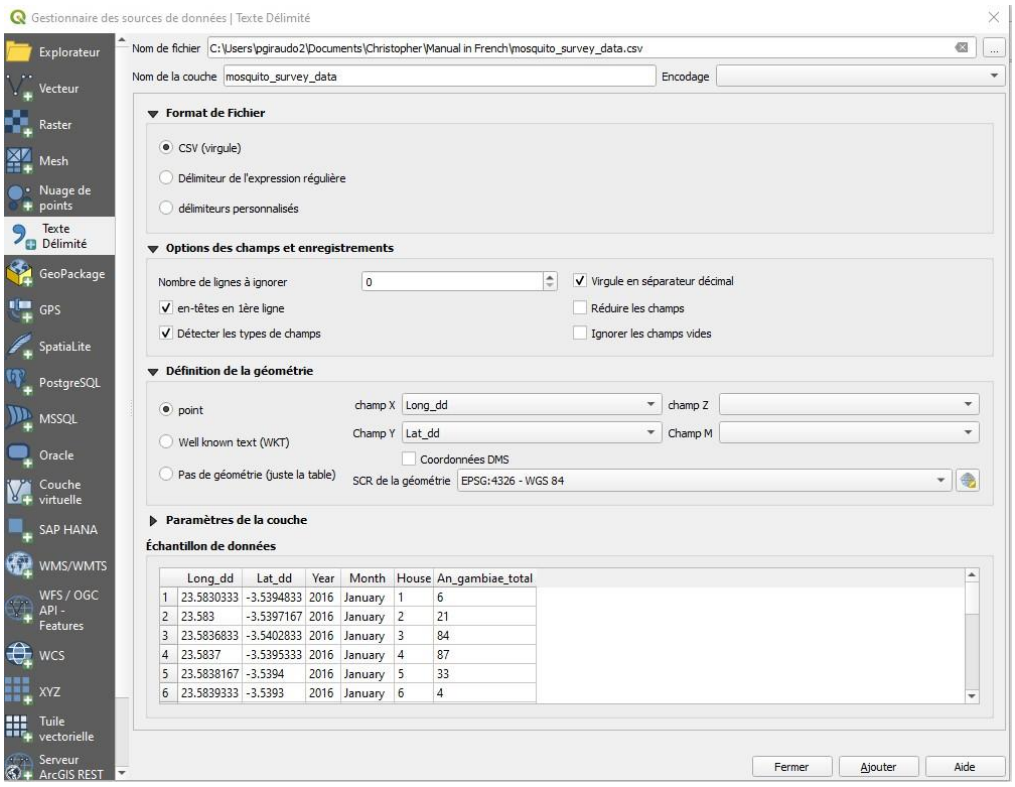


Figure 4.2. Fenêtre QGIS "Créer une couche à partir d'un fichier texte délimité".

Ensuite, nous vérifions que le système de référence des coordonnées (CRS) approprié est utilisé. Comme nos coordonnées sont ici en degrés décimaux, nous utilisons le CRS EPSG:4326 - WGS84. Vérifiez que ce système est sélectionné comme "SCR de

la géométrie". Si ce n'est pas le cas, définissez le système de référence de coordonnées approprié en cliquant sur  l'icône située à côté de la liste déroulante 'SCR de la géométrie'. Cela ouvre la fenêtre "Sélectionneur de système de coordonnées de référence " (Figure 4.3).

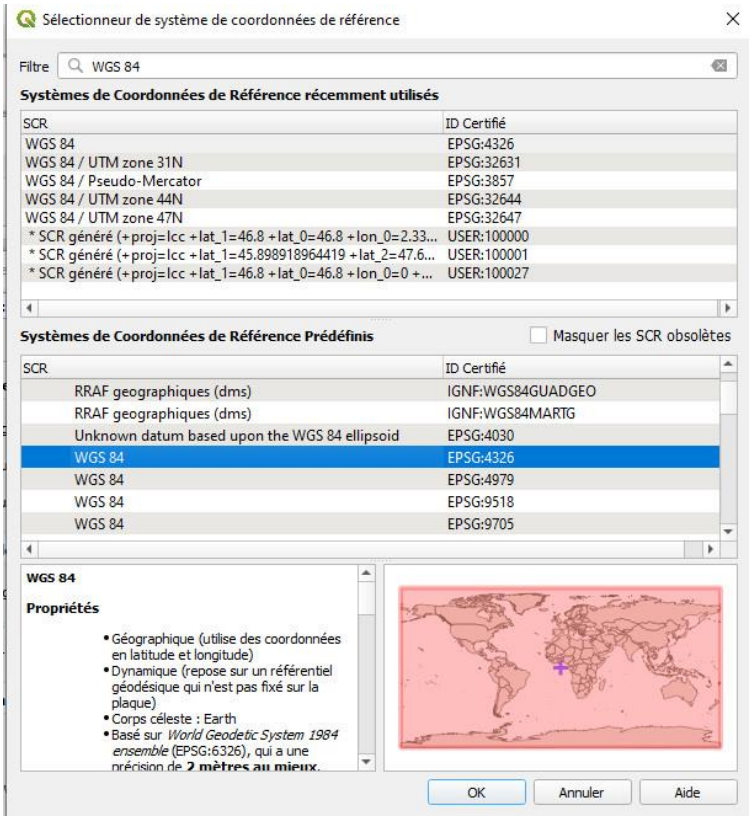


Figure 4.3. La fenêtre du sélecteur de système de référence de coordonnées.

- Ici, nous pouvons utiliser l'onglet de recherche "Filtre" en haut de la fenêtre pour trouver le système de coordonnées requis. Tapez 4326 dans le filtre de recherche, et l'option WGS84 EPSG:4326 apparaîtra sous la barre de filtre. Sélectionnez-la et cliquez sur 'OK'.
- De retour à la fenêtre "Data Source Manager", cliquez sur "Add" puis fermez, et les emplacements du fichier .csv devraient alors s'afficher (Figure 4.4). Le nom de la couche "mosquito_survey_data" doit apparaître dans le panneau "Couches".

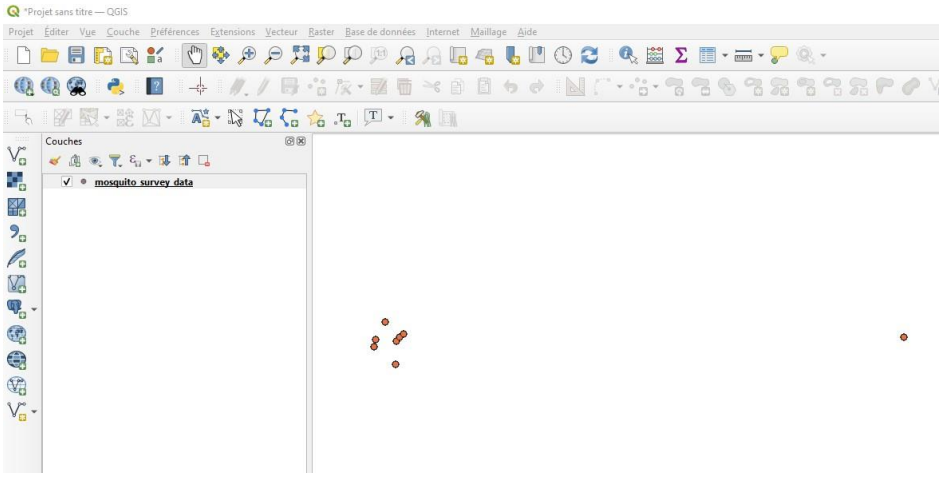


Figure 4.4. Les emplacements contenus dans le fichier .csv affichés dans QGIS.

- Pour enregistrer la couche mosquito_survey_data en tant que shapefile, cliquez avec le bouton droit de la souris sur la couche dans le panneau 'Couches', sélectionnez 'Exporter' puis 'Sauvegarder les entités sous...'. Spécifiez le format de fichier comme 'ESRI Shapefile' et ensuite spécifiez le nom de fichier du shapefile à sauvegarder (vous pouvez conserver le nom de fichier mosquito_survey_data), et spécifiez l'emplacement où le fichier doit être sauvegardé en utilisant l'icône. 

4.2 Création d'un shapefile de polygones d'entraînement

On génère maintenant une classification de l'occupation du sol de la zone d'étude. Selon la nature de la zone d'étude, les classes d'occupation du sol à cartographier peuvent différer. Dans cet exemple, une nomenclature de classification à huit classes sera utilisée, ces classes et les codes d'occupation du sol correspondants étant affichés dans le tableau 4.1. Ce guide de l'utilisateur est fourni avec des données d'entraînement et de validation pour la classification de l'exemple d'étude de Lodja, mais si les utilisateurs souhaitent créer leur propre classification pour la zone qui les intéresse, les sections 4.2 et 4.3 décrivent les étapes à suivre.

Tableau 4.1. Classes de couverture terrestre et codes correspondants.

Classe de couverture du sol	Code
Forêt	1
Pâturages	2
Défrichement	3
Jachère	4
Bâti	5
Eau courante	6
Eau stagnante	7
Brûlis	8

Idéalement, les données d'entraînement et de validation de la classification de l'occupation du sol sont basées sur des données collectées sur le terrain où les emplacements des types d'occupation du sol connus sont enregistrés avec un GPS. Ces données peuvent, le cas échéant, être complétées par des données d'entraînement/validation supplémentaires provenant d'images à très haute résolution (THR) de la zone d'intérêt, obtenues via des portails publics tels que Google Earth ou Bing Aerial. Les utilisateurs doivent être conscients des limites de l'utilisation de ces sources de données, notamment lorsqu'il existe un décalage temporel entre les dates d'acquisition des données satellitaires à classifier et l'imagerie THR utilisée comme référence. Nous ne proposons pas ici une revue de la méthodologie de collecte des données de classification ou d'entrainement/validation, mais nous encourageons les utilisateurs à consulter la documentation appropriée sur ces sujets afin d'améliorer leur compréhension du processus et de la façon dont les différentes approches peuvent avoir un impact sur la qualité des classifications.

L'ensemble de données d'entraînement pour la classification de l'occupation du sol qui sera effectuée dans Google Earth Engine sera un ensemble de données vectorielles ponctuelles contenant un nombre égal de points d'entraînement (emplacements) pour chaque classe d'occupation du sol. Initialement, un shapefile de polygones est créé avec plusieurs polygones correspondant à des zones de types d'occupation du sol connus. Ces polygones d'entraînement seront créés pour chaque classe d'occupation du sol à classer. Comme la distribution des classes d'occupation du sol dans une zone d'étude varie, certains types d'occupation du sol étant plus rares que d'autres, et comme la taille et le nombre de polygones créés pour chaque classe d'occupation du sol sont susceptibles de varier, nous créons ensuite un sous-ensemble de données d'entraînement à partir de ces polygones. Ce processus prend les polygones d'entraînement existants et sélectionne un nombre défini de points situés au hasard dans l'étendue spatiale des polygones pour chaque classe d'occupation du sol. Cela garantit qu'un ensemble équilibré de données d'entraînement est utilisé pour la classification, chaque classe d'occupation du sol étant également représentée dans les données d'entraînement. Le jeu de données de polygones d'entraînement sera importé dans GEE en tant que ressource, et le sous-échantillonnage et la conversion en un jeu de données ponctuelles seront effectués dans GEE.

Ici, les couches d'imagerie VHR de référence disponibles dans QGIS seront utilisées comme source de données de référence pour créer le shapefile initial d'entraînement sous forme de polygones.

- Ouvrez QGIS et installez l'extension 'QuickMapServices', qui contient un ensemble varié de sources de données, y compris l'imagerie VHR qui peut être utilisée comme carte de base pour la contextualisation et pour la collecte de données de référence. Pour ce faire, cliquez sur "Extensions" dans la barre d'outils principale, puis sur "Installer/gérer les extensions".
- La fenêtre pop-up 'Extensions' devrait s'ouvrir (Figure 4.5). Dans la barre de recherche, recherchez 'QuickMapServices', puis installez l'extension.

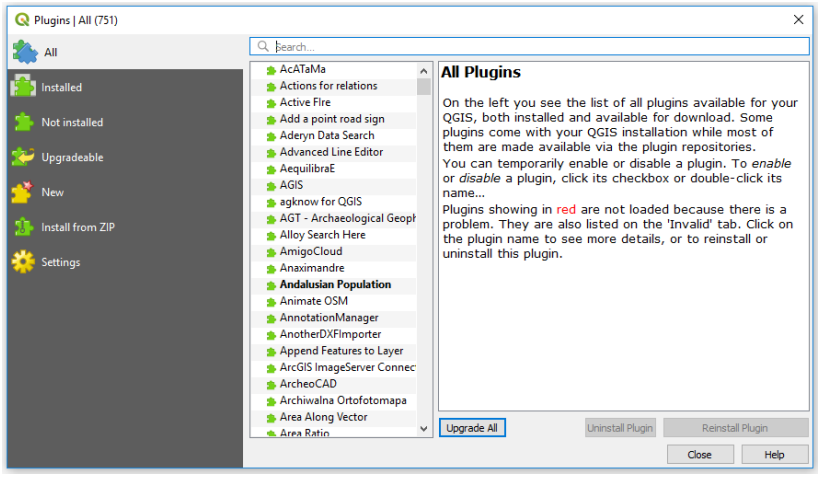


Figure 4.5. La fenêtre des plugins QGIS.

- Une fois installée, l'option QuickMapServices doit apparaître dans l'onglet "Internet" de la barre d'outils principale. Si vous la sélectionnez, elle affichera les couches qui sont actuellement disponibles pour être affichées comme cartes de base (Figure 4.6). Actuellement, cela ne donne qu'un nombre limité d'options - nous pouvons augmenter ces options en téléchargeant des modules supplémentaires.

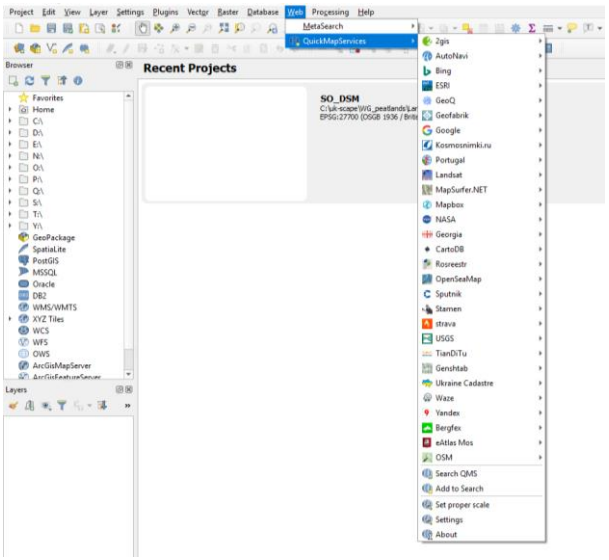


Figure 4.6 Options de la couche QGIS QuickMapServices.

- Pour ce faire, ouvrez l'option QuickMapServices dans la barre d'outils principale, puis cliquez sur l'option "Settings". Dans la boîte de dialogue qui s'ouvre, sélectionnez l'onglet "More services", puis cliquez sur "Get contributed pack". Cliquez sur "Enregistrer". Une gamme plus large de cartes de base et de produits d'imagerie devrait maintenant être disponible via QuickMapServices, y compris l'imagerie VHR des couches satellites de Google Satellite et Bing. Sélectionnez "Google Satellite" et une carte de base devrait apparaître, ainsi qu'une couche dans le "panneau des couches", qui peut être activée ou désactivée. Si aucune autre donnée n'est chargée dans la session QGIS, la carte de base du globe entier s'affichera initialement, mais vous pouvez utiliser

les fonctions de zoom pour vous concentrer sur la zone qui vous intéresse à un niveau de détail beaucoup plus élevé.

- Ensuite, créez un nouveau projet en cliquant sur l'icône située sous la barre d'outils principale. Ensuite, cliquez sur l'icône "Nouvelle couche shapefile"  sous la barre d'outils principale, et la fenêtre "Nouvelle couche shapefile" devrait s'ouvrir (Figure 4.7).

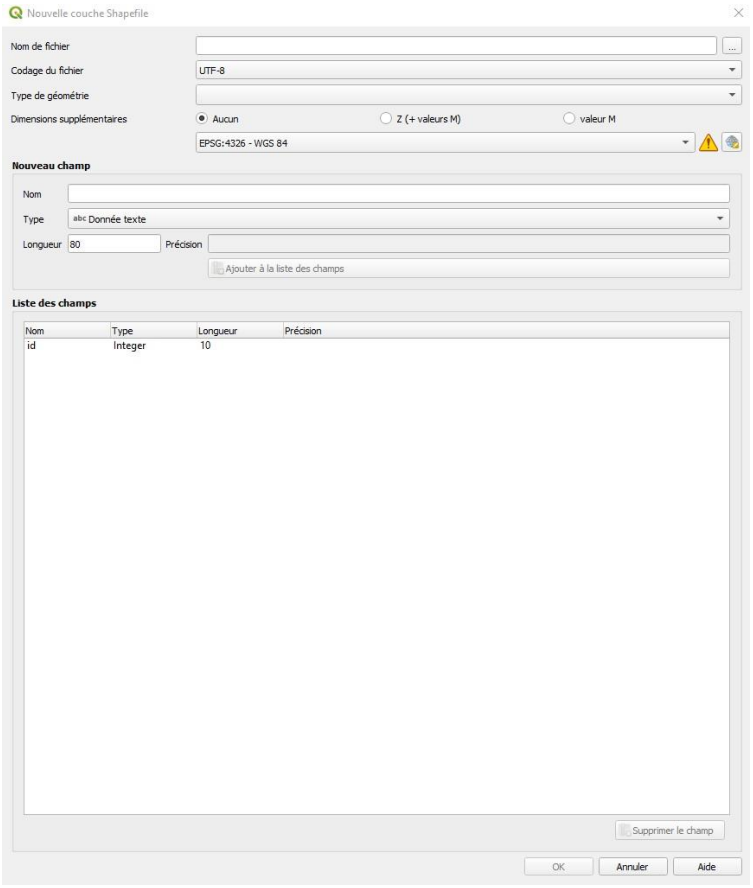


Figure 4.7. Fenêtre de nouveau shapefile.

- Remplissez les champs, en vous assurant que le "Type" de géométrie est un polygone et que vous sélectionnez le système de référence de coordonnées approprié (ici epsg :4326). Vous devez ensuite ajouter une colonne pour les classes de couverture du sol dans la section "Nouveau champ". Dans la case 'Nom', tapez 'Classe'. Assurez-vous que le menu déroulant 'Type' est réglé sur 'Nombre entier'. Cliquez sur "Ajouter à la liste des champs" pour ajouter ce champ au polygone.
- Enregistrez le polygone dans un répertoire approprié. Il devrait ensuite être ajouté en tant que couche dans l'onglet "Couches".
- Sélectionnez le shapefile, puis cliquez sur le bouton "Basculer en mode édition"  sur la barre d'outils du panneau supérieur, ou par le menu qui apparaît en clic droit. Sélectionnez ensuite le bouton "Ajouter une entité polygonale"  situé à proximité. .

- Dessinez autour d'une zone d'intérêt pour une classe de couverture terrestre particulière, en utilisant le bouton gauche de la souris pour ajouter des sommets et un clic droit pour fermer le polygone. Lorsque le polygone est terminé, la boîte "Attributs d'entités " demande que des détails soient ajoutés (Figure 4.8). Ajoutez l'identifiant du polygone et le numéro de la classe qui correspond à la classe d'occupation du sol en cours de numérisation.

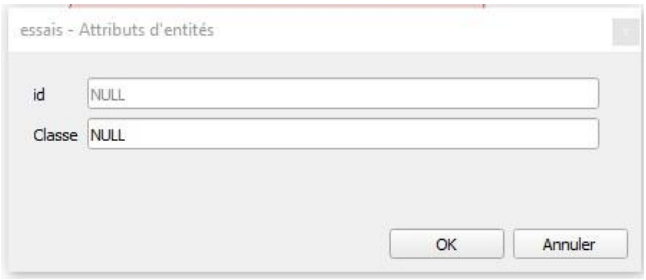


Figure 4.8. Fenêtre Attributs d'entité.

- Continuez à ajouter des polygones d'entraînement jusqu'à ce que l'ensemble de données d'entraînement soit complet.
- Pour terminer l'édition, enregistrez les polygones en cliquant sur le bouton "Enregistrer"  , puis terminez l'édition en cliquant sur le bouton "Modifier"  . Il est recommandé de sauvegarder vos modifications à intervalles réguliers tout au long de ce processus.

4.3 Pour créer le shapefile de l'étendue de la zone d'étude

Nous devons également créer un fichier de forme délimitant la zone d'intérêt de l'étude afin de limiter la zone des données traitées dans le moteur Google Earth et exportées depuis celui-ci. Pour ce faire, suivez les étapes décrites dans la section "4.2 Création d'un fichier de forme de polygone d'entraînement" pour créer un nouveau fichier de forme. Ajoutez ensuite un polygone unique correspondant à la zone d'intérêt et enregistrez le polygone. Il n'est pas nécessaire d'ajouter d'autres attributs à ce fichier de forme, mais vous pouvez le faire si vous le souhaitez.

5 Google Earth Engine - prétraitement des données satellitaires

5.1 Interface GEE / GUI

Google Earth Engine (GEE) est la plateforme de traitement sur le cloud qui permet de réaliser une grande partie du prétraitement des données avant l'étape suivante de modélisation. Avant de commencer l'analyse, nous allons nous familiariser avec l'interface de GEE.

Tout d'abord, ouvrez une session dans un navigateur (par exemple Google Chrome) et accédez à https://code.earthengine.google.com. L'interface GEE apparaît comme à la Figure 5.1, avec ici un exemple de script et le résultat de l'analyse affichés.

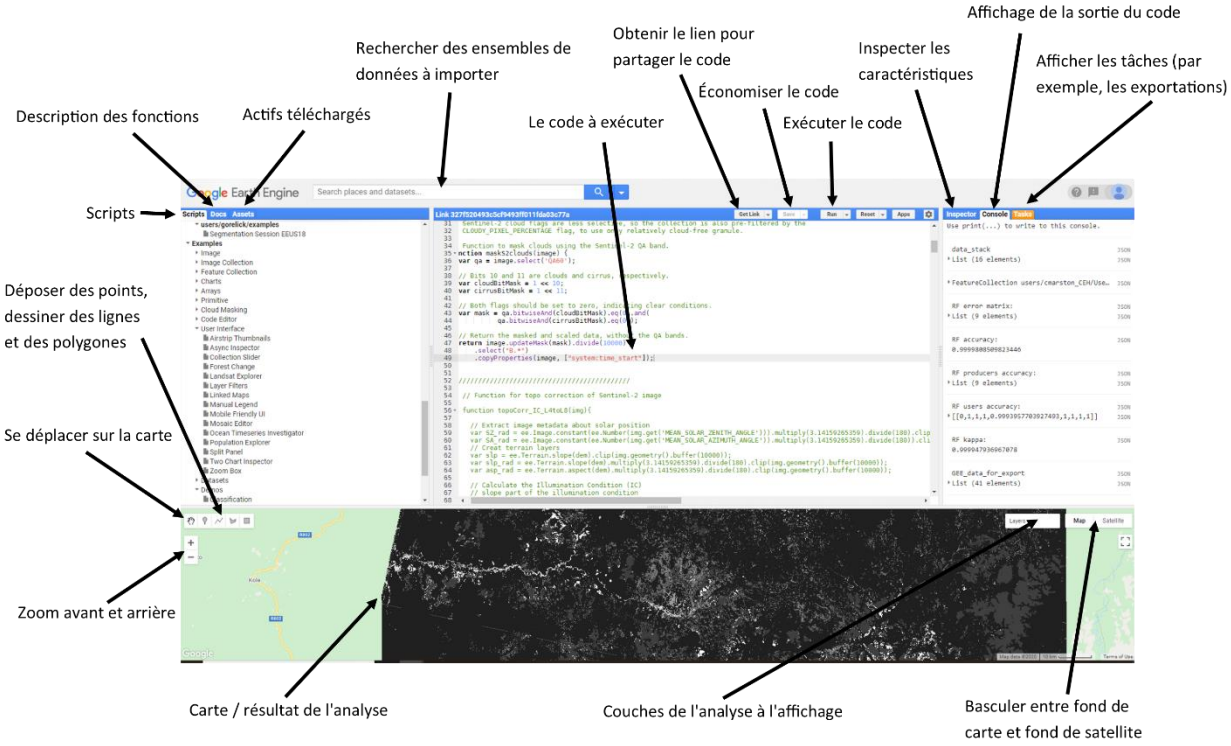


Figure 5.1. Interface du moteur Google Earth.

GEE implémente ses fonctionnalités via des lignes de commande en utilisant le langage de programmation Javascript. Des commandes spécifiques sont donc passées sous forme de lignes de code correspondant à différentes applications de traitement de données. Plusieurs commandes peuvent être reliées entre elles pour des tâches plus complexes de traitement des données en plusieurs étapes, les scripts pouvant être partagés avec d'autres utilisateurs. Il est vivement conseillé de sauvegarder les révisions apportées aux scripts à intervalles réguliers. Si vous souhaitez partager les scripts que vous avez développés avec d'autres utilisateurs, vous pouvez le faire en cliquant sur "Obtenir le lien", puis dans la fenêtre qui s'ouvre, cliquez sur l'icône "Cliquer pour copier le lien". Vous pouvez ensuite distribuer le lien comme il convient

aux autres utilisateurs qui pourront accéder au script. Si le script repose sur des ressources qui ont été téléchargées sur un compte utilisateur, il sera nécessaire de partager ces ressources. Pour ce faire, naviguez jusqu'à l'emplacement approprié dans l'onglet "Assets", passez le curseur sur le nom de fichier de la ressource appropriée et cliquez sur l'icône "partager". Dans la fenêtre qui s'ouvre, cochez la case "Tout le monde peut lire", puis cliquez sur "Terminé". La ressource sera alors accessible aux autres utilisateurs lorsqu'ils exécuteront le script que vous avez distribué.

Pour ouvrir un script GEE qui a été partagé avec vous, il suffit de coller le lien dans une session de navigateur ou de double-cliquer sur l'hyperlien. Vous accéderez alors à une session d'éditeur de code GEE affichant le script que vous pourrez ensuite exécuter.

Bien que ce tutoriel utilise un script développé pour exécuter une analyse similaire à celle contenue dans Marston *et al.* (2023), un éventail beaucoup plus large de fonctionnalités et d'ensembles de données est disponible dans GEE et les utilisateurs sont encouragés à les explorer via le site Web de GEE à l'adresse https://earthengine.google.com/.

5.2 Téléchargement des ressources

Certains jeux de données qui serviront à modéliser l'abondance des moustiques peuvent être directement exportés de GEE, mais d'autres (comme la proportion des couvertures au sol et la distance à la parcelle la plus proche d'un type de couverture terrestre donné) seront générés par l'utilisateur dans GEE. Pour ce faire, il est nécessaire de télécharger des jeux de données supplémentaires dans GEE pour effectuer et valider les classifications de couverture du sol. Ces ensembles de données doivent être téléchargés en tant que ressources et comprendront le shapefile délimitant la zone d'intérêt (AOI, *Area Of Interest*) et celui des zones d'entraînement des types d'occupation du sol, et celui des emplacements de validation pour évaluer la précision des classifications de l'occupation du sol. Les points et/ou les polygones peuvent être ajoutés directement dans GEE, cependant, comme les données telles que celles-ci sont souvent collectées à l'avance pendant les campagnes de travail sur le terrain et sont souvent nombreuses, une façon plus pratique d'ajouter ces données dans GEE est de charger un shapefile existant contenant les données requises comme une ressource. Pour charger une ressource, dans la session GEE, sélectionnez l'onglet 'Ressources' (Assets) (en haut à gauche), cliquez sur 'Nouveau' puis 'Shapefiles'. Cliquez sur "Select", naviguez jusqu'à l'emplacement où le shapefile est enregistré. Bien que les shapefiles soient considérés comme un seul fichier dans un SIG, ils sont en fait composés de plusieurs fichiers qui auront le même préfixe (nom de fichier), mais différentes extensions de fichier, par exemple .shp, .shx, .dbf, .prj et autres. Ces multiples fichiers fonctionnent en combinaison pour définir la géométrie et les attributs des données à afficher ou à analyser. Lors du téléchargement d'un shapefile vers GEE, tous les fichiers constitutifs du shapefile doivent être sélectionnés, à l'exception du fichier .sbx. Si le fichier .sbx est sélectionné, le processus de téléchargement échouera. Si cela se produit, essayez de télécharger à nouveau le shapefile sans inclure ce fichier qui cause le problème.

À cette occasion, nous aurons besoin de données d'entraînement identifiant les emplacements de types de couverture terrestre connus pour effectuer une classification de la couverture terrestre, et d'un ensemble de données de validation

distinct pour effectuer une évaluation de la précision de la classification. Celles-ci ont déjà été téléchargées en tant que deux ressources distinctes et partagées, les données d'entraînement en tant que shapefile de polygones, et les données de validation en tant que shapefile de points.

5.3 Exécution du script GEE

Trois scripts Google Earth Engine sont fournis avec ce manuel d'utilisation et sont utilisés pour les étapes séquentielles suivantes ;

1. Préparation des données
2. Modélisation des données
3. Visualisation des données

Les scripts sont disponibles via les hyperliens fournis dans les sections correspondantes de ce manuel d'utilisation. Si vous utilisez la touche Ctrl et cliquez sur ce lien, une session de l'éditeur de code GEE contenant le script s'ouvrira. Vous pouvez également copier et coller ce lien dans une session de navigateur Web.

Script 1 : Pré-traitement des données

https://code.earthengine.google.com/4d0a986266711f5490062201fd35ef45

Le code apparaîtra comme celui de la figure 5.2.

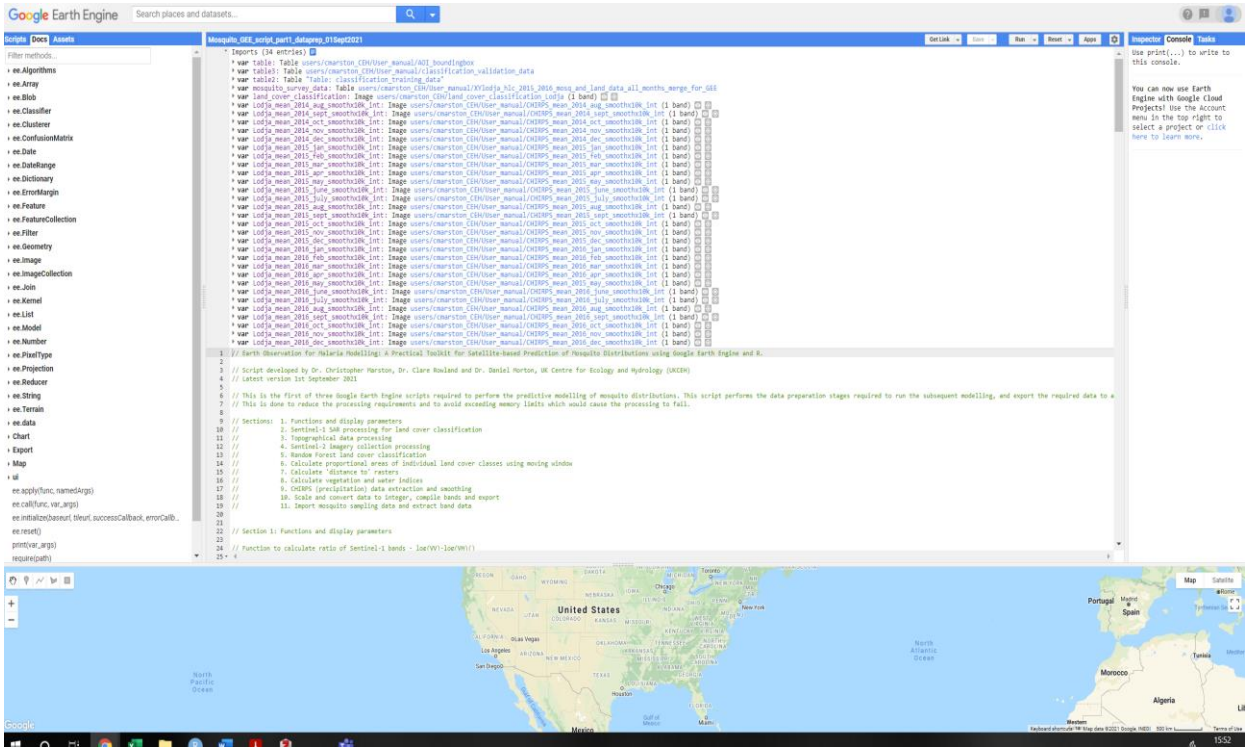


Figure 5.2. L'éditeur de code de Google Earth Engine avec le script affiché.

Les sections suivantes fournissent une explication des différentes étapes du traitement des données et de la modélisation. Les instructions sur l'exécution des scripts se trouvent à la section 5.6.

Le traitement des données GEE nécessite plusieurs étapes, qui sont décrites ci-dessous. Le code requis pour exécuter ces étapes est décrit ci-dessous. Notez que lorsqu'un exemple est donné (par exemple pour effectuer une étape d'analyse des données pour une classe de couverture du sol particulière), il peut être nécessaire de le répéter pour d'autres variables d'intérêt (par exemple, d'autres classes de couverture du sol). Pour des raisons de brièveté, un seul exemple est fourni et décrit dans ce document, mais le script GEE qui l'accompagne contient le code complet. Le script GEE est divisé en sections comme suit :

- Fonctions et paramètres d'affichage
- Traitement du SAR Sentinel-1 pour la classification de la couverture terrestre
- Traitement des données topographiques
- Traitement de la collecte de l'imagerie Sentinel-2
- Classification de l'occupation du sol par Random Forest
- Calcul des proportions en surface des différentes classes d'occupation du sol en utilisant une fenêtre mobile.
- Calcul de la "distance aux" rasters
- Calcul des indices de végétation et d'eau
- Extraction et lissage des données CHIRPS (précipitations)
- Mise à l'échelle et conversion des données en nombre entier, compilation des bandes et exportation.
- Import des données d'échantillonnage des moustiques et extraction des données des bandes.

Les sections du script indiquées par // sont "commentées", elles sont reconnues par GEE comme étant du code inerte, c'est-à-dire du texte descriptif et non une commande à exécuter (par exemple, Figure 5.3). Ceci est utile pour ajouter des commentaires ou des notes décrivant ce que font les commandes à certaines étapes du script, ou pour désactiver des commandes sans les supprimer du script lorsque cela s'avère utile. Les sections du script qui sont commentées sont affichées en vert.

```
22  // Section 1: Functions and display parameters
23
24  // Function to calculate ratio of Sentinel-1 bands - log(VV)-log(VH)()
25  var vh_vv = function(image) {
26    return image.select('VH').divide(image.select('VV'));
27  };
```

Figure 5.3. Exemple de script avec '//' et le texte vert indiquant les commentaires

Bien que le contenu de ce manuel d'utilisation et le code qu'il contient soient basés sur la recherche contenue dans Marston *et al.* (2023), il existe de légères différences dans les méthodes et la mise en œuvre entre cet article et ce guide d'utilisation. Ces adaptations sont délibérées et conçues pour utiliser l'abondance plus riche de données d'observation de la Terre par satellite disponibles depuis la période de collecte de données sur le terrain impliquée dans Marston *et al.* (2023), en vue de la mise en œuvre future de ces méthodes dans d'autres régions endémiques du paludisme. Principalement, alors que Marston *et al.* (2023) utilisaient des images uniques sans nuage, les méthodes présentées dans ce guide de l'utilisateur utilisent plutôt des collections d'images en combinaison pour générer des composites sans nuage, offrant une meilleure flexibilité et une opportunité d'application pratique de ces méthodes dans des zones où les opportunités d'obtenir des images uniques sans nuage sont faibles.

5.4 Importation des ressources (assets)

La section 5.2 décrit comment vous pouvez télécharger des ressources pour analyse dans GEE si vous avez besoin de le faire pour vos propres sites d'étude et applications. Pour visualiser les ressources qui ont été téléchargées et sont disponibles, cliquez sur l'onglet "Assets" en haut à gauche de la fenêtre GEE (Figure 5.4). Cela affichera les ressources disponibles pour l'analyse.

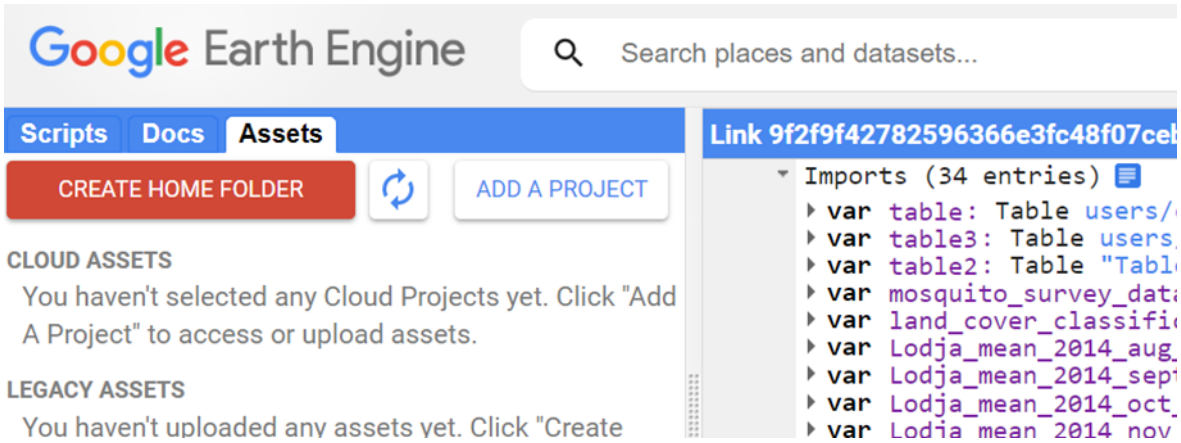


Figure 5.4. Onglet « Assets » de la console GEE

Si vos ressources sont enregistrées dans des sous-répertoires, vous pouvez développer le sous-répertoire pour afficher son contenu. Lorsque le curseur est placé sur la ressource qui vous intéresse, trois icônes apparaissent : "Share" (partager), "Rename" (Renommer) et "Import into script " (Importer dans le script). Pour rendre une ressource disponible pour l'analyse à l'aide du script GEE, vous devez l'importer en cliquant sur l'icône ""Import into script ". Une fois importé, la ressource apparaîtra en haut de la fenêtre de code et sera disponible en tant que ressource pouvant être appelé par le code. Le nom de la ressource est également indiqué, par exemple dans la Figure 5.5, les trois premières ressources importées ont reçu les noms table, table2 et table3.

Figure 5.5. Ressources importées dans le script GEE.

Pour les besoins de cet exercice de formation, les actifs requis ont déjà été téléchargés et sont lus au début du script GEE correspondant.

5.5 Description du script

5.5.1 Fonctions et paramètres d'affichage

Pour exécuter une commande particulière sur chaque élément d'une collection, nous définissons d'abord l'opération que nous souhaitons appliquer à chaque élément de la collection sous forme de fonction. Nous sommes ensuite en mesure d'appliquer (on parle de mapper), cette fonction sur chaque objet ou image dans une collection spécifiée en utilisant la commande `map()`. Par exemple, si la fonction consiste à générer une couche d'indice de végétation par différence normalisée (NDVI) pour une image particulière, et que la collection contient toutes les images Sentinel-2 pour un lieu donné sur une année, le mappage de la fonction sur la collection générera des couches NDVI pour chaque image de la collection.

Dans cet exemple, les fonctions utilisées dans le script ne nécessitent pas d'édition. Elles ne sont donc pas examinées en détail ici, si ce n'est pour énumérer ce que font les fonctions :

- Calcul d'une couche du rapport VV/VH à partir des bandes de polarisation VV et VH de Sentinel-1 ;
- Masquage des nuages de l'imagerie Sentinel-2 et des bords de l'image.
- Calcul de l'indice de végétation par différence normalisée (NDVI).

- Calcul de l'indice de végétation ajusté au sol (SAVI)
- Calcul de l'indice de teneur en eau par différence normalisée (NDWI)
- Calcul de l'indice de teneur en eau par différence normalisée modifié (MNDWI)

Ces fonctions seront mises en correspondance (mappées) avec les collections de données aux étapes appropriées du script.

Ensuite, nous prédéfinissons une palette de couleurs qui peut être utilisée pour afficher la classification de l'occupation du sol que nous allons générer pour examen. Une couleur spécifique est définie pour chaque classe qui sera cartographiée à l'aide du code ci-dessous. La commande `var` demande à GEE de créer un nouvel objet, et ceci suivi du nom de l'objet à créer que nous appellerons ici `palette`. Le code suivant spécifie ensuite ce que sera l'objet créé - ici il s'agira d'une série de codes hexadécimaux correspondant chacun à une couleur. Comme il y aura huit classes d'occupation du sol dans la classification, huit codes hex sont spécifiés. Chaque code est ensuite suivi d'un texte commenté indiquant à quelle classe la couleur spécifique s'applique, quelle est cette classe et la couleur qui sera affichée. Comme ce texte est commenté, il ne s'agit pas de commandes exécutées par GEE, mais d'une note de référence utile pour l'utilisateur. Vous trouverez plus de détails sur les codes hexadécimaux et les couleurs correspondantes à l'adresse https://cloford.com/resources/colours/500col.htm.

```
var palette = [
  '008000', // Class 1, forest, green
  'FFFF00', // Class 2, grassland, yellow
  'C0C0C0', // Class 3, clearing, silver
  '800000', // Class 4, fallow, maroon
  '000000', // Class 5, built-up, black
  '00FFFF', // Class 6, flowing water, cyan
  '0000FF', // Class 7, static water, blue
  '800080', // Class 8, burnt, purple
];
```

Ensuite, nous spécifions la distance de vol connue de l'espèce de moustique que nous modélisons ici, *Anopheles gambiae*. Celle-ci est spécifiée comme étant de 846m dans Verdonschot et Besse-Lototskaya (2014).

```
var flight_distance = 846;
```

Après avoir défini les fonctions, les paramètres d'affichage et la distance de vol qui sont nécessaires dans le script GEE, nous passons aux étapes de préparation des données.

5.5.2 Traitement du SAR Sentinel-1 pour la classification de l'occupation du sol

La série de commandes suivante sélectionne une collection d'images SAR (Synthetic Aperture Radar) de Sentinel-1, extrait les données de polarisation VV et VH de cette collection, calcule un produit de données (rapport VV/VH) et génère des ensembles de données de valeur médiane de pixel pour chacun des produits de données VV, VH et de rapport. Tout d'abord, nous créons la collection d'images Sentinel-1.

```
// Get the Sentinel-1 collection.
var collectionS1 = ee.ImageCollection('COPERNICUS/S1_GRD')
.filter(ee.Filter.eq('instrumentMode', 'IW'))
.filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VV'))
.filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VH'))
.filter(ee.Filter.eq('orbitProperties_pass', 'DESCENDING'))
.filterBounds(table)
.filterDate('2015-01-01', '2016-12-31')
```

Décomposons ce que font ces commandes ligne par ligne. La première ligne commentée identifie simplement pour l'utilisateur les actions que le bloc de commandes suivant exécute.

```
// Get the Sentinel-1 collection.
```

La ligne suivante demande à GEE de créer une collection d'images SAR Sentinel-1. `var` indique qu'un nouvel objet est créé par la ou les commandes suivantes. `collectionS1` est le nom de l'objet (la collection d'images) qui est créé. `=` indique que la collection est créée sur la base de la (des) commande(s) suivante(s). `ee.ImageCollection('COPERNICUS/S1_GRD')` indique que la collection d'images que nous créons s'appuiera sur les archives intégrées dans GEE de l'imagerie SAR Sentinel-1 dont l'identifiant de jeu de données GEE est 'COPERNICUS/S1_GRD'. Un large éventail de jeux de données pré-intégrés est disponible dans GEE, chacun ayant son propre identifiant. Si nous souhaitions exécuter cette fonction sur un autre jeu de données, nous devrions modifier la mise à jour de l'identifiant du jeu de données dans cette commande pour qu'il corresponde à l'identifiant du jeu de données alternatif que nous souhaitons utiliser. De plus amples informations sur les différents ensembles de données disponibles dans GEE et leurs identifiants respectifs sont disponibles sur le site Web de GEE (https://earthengine.google.com/) si nécessaire.

```
var collectionS1 = ee.ImageCollection('COPERNICUS/S1_GRD')
```

Actuellement, la commande crée un objet contenant toutes les images Sentinel-1 disponibles dans les archives de GEE. Nous devons ensuite effectuer une série

d'opérations de filtrage pour ne sélectionner que les images correspondant aux produits de données, à la zone géographique et aux périodes qui nous intéressent. Tout d'abord, nous spécifions que nous n'avons besoin que du mode d'acquisition de données Interferometric Wide Swath. Sentinel-1 acquiert des données dans plusieurs modes d'acquisition de données, mais ici nous ne sommes intéressés que par le mode Interferometric Wide Swath data acquisition (interférométrique à large bande). Nous n'entrerons pas ici dans les détails des différents modes d'acquisition de données et des caractéristiques des données de Sentinel-1, mais nous encourageons l'utilisateur à se familiariser avec la littérature sur ce sujet. La ligne suivante spécifie que la collection sera filtrée pour ne retenir que le mode instrument = Interferometric Wide Swath, identifié dans le code comme IW.

```
.filter(ee.Filter.eq('instrumentMode', 'IW'))
```

Les données SAR de Sentinel-1 sont acquises dans deux polarisations, VV et VH. Nous avons besoin de ces deux polarisations pour l'analyse ultérieure des données et nous effectuons ici des étapes de filtrage supplémentaires pour ne retenir que les images qui contiennent d'abord les polarisations VV, puis les polarisations VH. Nous filtrons également la collection pour retenir les images acquises uniquement sur une trajectoire orbitale descendante.

```
.filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VV'))
.filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VH'))
.filter(ee.Filter.eq('orbitProperties_pass', 'DESCENDING'))
```

Ensuite, nous allons filtrer davantage la collection pour retenir les images qui couvrent notre zone d'intérêt. La zone d'intérêt est ici définie par la table des objets, qui est le shapefile de délimitation (bounding box) de la zone d'intérêt. Elle a été importée en tant que ressource à un stade antérieur. Cette opération ne conservera que les images qui sont incluses dans cette zone. Notez que certaines des images retenues peuvent ne pas couvrir toute l'étendue de la zone d'intérêt, et la couverture de certaines images peut s'étendre au-delà de la limite de la zone d'intérêt.

```
.filterBounds(table)
```

Enfin, nous allons filtrer pour ne retenir que les images acquises pendant la période qui nous intéresse. Pour cela, il nous suffit de spécifier dans un premier temps les dates de début et dans un second temps les dates de fin de la période qui nous intéresse. Ici, nous avons spécifié une période allant du 1er janvier 2015 au 31 décembre 2016. Pour cette commande, les dates sont saisies au format YYYY-MM-JJ.

```
.filterDate('2015-01-01', '2016-12-31')
```

Les commandes de filtrage seront exécutées séquentiellement pour produire la collection d'images relatives à notre emplacement, la période de temps et les caractéristiques de données d'intérêt. Ensuite, nous prendrons cette collection et effectuerons d'autres étapes d'analyse sur celle-ci.

La collection d'images contiendra un grand nombre d'images acquises à des dates différentes. Pour certaines applications, cependant, nous ne voulons pas un grand nombre d'images mais une seule image qui représente les caractéristiques "typiques" de la zone d'intérêt sur l'ensemble de la période, plutôt que sur un seul instantané. GEE utilise des "réducteurs" pour convertir une collection d'images en une seule image de sortie basée sur des paramètres définis par l'utilisateur. Nous n'allons pas explorer les réducteurs en profondeur ici, mais nous allons appliquer à la collection de données un réducteur calculant la médiane des valeurs de chaque pixel. Pour un emplacement donné, il calculera la valeur médiane de tous les pixels non masqués de la collection pour un emplacement de pixel. Cette valeur médiane forme la valeur du pixel pour cet emplacement dans l'image de sortie. Cette opération est répétée pour tous les pixels de la collection d'images, produisant ainsi la bande matricielle (raster) de sortie.

Nous générons la couche médiane à l'aide du code ci-dessous. Nous combinons ici deux commandes dans la même ligne de code, les commandes étant appliquées séquentiellement dans l'ordre où elles sont écrites pour créer une nouvelle image nommée VVonly_med contenant les valeurs médianes des pixels des données de polarisation VV de la collection Sentinel-1 précédemment créée. La première commande sélectionne uniquement les données de polarisation VV de la collectionS1. La deuxième commande crée une nouvelle image qui contient les valeurs médianes des pixels VV. Notez que cela crée une image individuelle, plutôt qu'une collection d'images comme nous l'avons fait précédemment.

```
var VVonly_med = collectionS1.select('VV').median();
```

Ensuite, nous répétons ce processus pour la polarisation VH.

```
var VHonly_med = collectionS1.select('VH').median();
```

Enfin, nous calculons la bande du rapport VV/VH. Nous ne pouvons pas la sélectionner de la même manière que pour les bandes VV et VH car la bande du rapport n'existe pas encore - nous devons la calculer. Pour ce faire, nous devons mapper la fonction vh_vv qui a été définie sur la collectionS1, puis appliquer un réducteur médian.

```
var ratio_med = collectionS1.map(vh_vv).median();
```

La fonction est mappée ici en utilisant la commande collectionS1.map(vh_vv) où collectionS1 est la collection sur laquelle la fonction sera mappée, map() est la commande pour mapper la fonction, et vh_vv est le nom de la fonction qui est mappée. Nous appliquons ensuite un réducteur « médiane » pour générer une seule couche avec la médiane du

rapport VV/VH pour chaque pixel, de façon analogue à ce nous avons fait précédemment pour les couches VV et VH prises séparément.

5.5.3 Traitement des données topographiques

Les données topographiques sont utilisées dans cette analyse de deux manières : pour être incluses dans l'étape de classification de l'occupation du sol, et aussi pour être analysées comme variables indépendantes en relation avec l'abondance des moustiques. Nous sommes intéressés par quatre produits de données topographiques, à savoir l'élévation, l'orientation, la pente et l'indice de position topographique (TPI). Tous ces produits utilisent les données du modèle numérique de terrain (DEM, digital elevation mondel) de la Shuttle Radar Topography Mission (SRTM) déjà intégrées dans GEE. Tout d'abord, nous utilisons les données SRTM DEM, ici désignées par 'USGS/SRTMGL1_003' pour créer une nouvelle image appelée dataset. Notez que comme nous utilisons un seul jeu de données DEM plutôt qu'un certain nombre d'images satellites différentes comme nous l'avons fait précédemment, nous créons un objet image plutôt qu'une collection d'images.

```
var dataset = ee.Image('USGS/SRTMGL1_003');
```

Nous sélectionnons ensuite l'élévation dans le jeu de données et l'utilisons pour créer une nouvelle image elevation_int. Nous utilisons la commande .toInt() pour convertir l'image en valeurs entières.

```
var elevation_int = dataset.select('elevation').toInt();
```

A partir de l'image elevation_int, nous calculons ensuite la pente et l'orientation et les sauvegardons sous forme d'images séparées. Une fois encore, les images de pente et de l'élévation sont converties en valeurs entières, mais comme la conversion directe des pentes décimales en valeurs entières entraînerait une perte de détails dans ces données, nous multiplions d'abord les valeurs de pente par 10000.

```
var slopex10k_int= ee.Terrain.slope(elevation_int).multiply(10000).toInt();
var aspect_int= ee.Terrain.aspect(elevation_int).toInt();
```

Ensuite, nous calculons l'indice de position topographique (TPI). Pour ce faire, nous calculons l'élévation de chaque pixel et la soustrayons de l'élévation moyenne de la zone environnante, dans ce cas sur une zone circulaire de 15 pixels de rayon. Cela crée l'image tpi_15_pixel_int, où elevation est le DEM spécifié précédemment et 15 est le nombre de pixels défini comme le rayon de la zone circulaire sur laquelle la valeur d'élévation moyenne sera calculée. Nous appliquons ensuite la commande .toInt() pour spécifier que l'ensemble de données généré sera au format entier.

```
var tpi_15_pixel_int = elevation_int.subtract(elevation_int.focal_mean(15)).toInt();
```

5.5.4 Traitement de la collection de l'imagerie Sentinel-2

Ensuite, nous créons une nouvelle collection d'images pour les images optiques Sentinel-2 qui seront utilisées en combinaison avec les ensembles de données Sentinel-1 et topographiques pour effectuer la classification de l'occupation du sol, et pour générer des produits de données d'indice de végétation. Les commandes ci-dessous créent une collection d'images en sélectionnant les données de l'archive de données de réflectance de surface Sentinel-2 (identifiée dans GEE sous le nom de 'COPERNICUS/S2_HARMONIZED'), et filtrent la collection en fonction de l'étendue de la zone d'étude telle que spécifiée dans la ressource 'table', et la plage de dates spécifiée. Les données optiques de Sentinel-2, contrairement aux données SAR de Sentinel-1, sont affectées par les nuages. Cela pose un problème dans de nombreuses régions du monde, notamment dans les régions où le paludisme est endémique. Heureusement, la fonctionnalité de GEE permet de générer une nouvelle image composite à partir d'une série d'images qui peuvent être partiellement affectées par les nuages, augmentant ainsi la zone de couverture sans nuage pour l'analyse. Le masquage des nuages de l'imagerie Sentinel-2 est également effectué à l'aide du produit Sentinel-2 Cloud Probability. Tout d'abord, nous spécifions les produits de données d'imagerie et de probabilité des nuages que nous souhaitons utiliser, et définissons une valeur seuil maximale de probabilité des nuages, ici 65, que nous souhaitons appliquer.

```
var s2Sr = ee.ImageCollection('COPERNICUS/S2_HARMONIZED');
var s2Clouds = ee.ImageCollection('COPERNICUS/S2_CLOUD_PROBABILITY');
var MAX_CLOUD_PROBABILITY = 65;
```

Nous spécifions ensuite l'étendue `.filterBounds` et la plage de dates qui seront appliquées pour filtrer les deux ensembles de données.

```
var criteriaS2 = ee.Filter.and(
   ee.Filter.bounds(table), ee.Filter.date('2015-01-01', '2016-12-31'));
```

Nous appliquons ensuite ces critères de filtrage aux données `S2_SR` et Cloud Probability que nous souhaitons utiliser. Pour la collection `S2_SR`, nous mappons également la fonction `maskEdges()` ici.

```
var s2Sr = s2Sr.filter(criteriaS2).map(maskEdges);
var s2Clouds = s2Clouds.filter(criteriaS2);
```

Ensuite, nous joignons la collection S2_SR à l'ensemble de données Cloud Probability pour ajouter le masque de nuage, créant ainsi la nouvelle collection d'images s2SrWithCloudMask.

```
var s2SrWithCloudMask = ee.Join.saveFirst('cloud_mask').apply({
  primary: s2Sr,
  secondary: s2Clouds,
  condition:
    ee.Filter.equals({leftField: 'system:index', rightField: 'system:index'})
});
```

Ensuite, nous créons une nouvelle collection d'images à partir de la collection s2SrWithCloudMask en appliquant la fonction maskClouds(), et en conservant uniquement les bandes spectrales de ces images dont nous avons besoin pour la classification de l'occupation du sol. Ici, nous retenons les bandes 2, 3, 4, 5, 6, 7, 8, 8A, 11 et 12. Les bandes 1, 9, 10 et 11 sont optimisées pour les applications atmosphériques, qui ne sont pas pertinentes dans le contexte de ce travail et ne sont donc pas prises en compte ici.

```
var S2_bandsubset =
ee.ImageCollection(s2SrWithCloudMask).map(maskClouds).select('B2','B3','B4','B5','B6','B7','B8','B8A','B
11','B12');
```

Nous appliquons ensuite un réducteur 'médiane' à la collection d'images S2_bandsubset pour créer l'image des médianes S2_med_bandsubset .

```
var S2_med_bandsubset = S2_bandsubset.median();
```

Enfin, nous combinons les ensembles de données Sentinel-2, Sentinel-1 et topographiques en une seule image multibande sur laquelle nous effectuerons la classification de l'occupation du sol. La nouvelle image que nous allons créer sera appelée data_stack, avec l'image S2_med_bandsubset formant les premières bandes. À cela, nous ajoutons les bandes VVonly_med, VHonly_med, ratio_med, elevation_int, slopex10k_int et aspect_int dans cet ordre.

```
var data_stack =
S2_med_bandsubset.addBands(VVonly_med).addBands(VHonly_med).addBands(ratio_med).addBands(
elevation_int).addBands(slopex10k_int).addBands(aspect_int);
```

5.5.5 Classification de la couverture terrestre par random forest

Maintenant que nous avons créé la pile de données, nous pouvons effectuer la **classification de l'occupation du sol**. Tout d'abord, nous créons une nouvelle collection de caractéristiques appelée polygones à partir de la ressource table2 que nous avons importé précédemment. Elle contient une série de polygones correspondant à des zones de types d'occupation du sol connus qui seront utilisés pour entraîner le classificateur. Chaque polygone contient l'attribut 'classcode' qui est un code entier où chaque valeur correspond à un type de couverture du sol différent (Tableau 1).

```
var polygons = table2;
```

Ensuite, nous extrayons les valeurs de pixel pour chaque bande de la pile de données pour les emplacements des polygones d'entraînement. Dans les commandes suivantes, classification_training est le nom de l'objet contenant les données d'entraînement que nous allons créer, data_stack est le nom de la pile de données d'imagerie d'où sont extraites les valeurs de pixels, polygons correspond à la collection de caractéristiques des polygones d'entraînement, et scale : 10, et spécifie une résolution spatiale de 10m. tileScale : 16 est un paramètre relatif au traitement de fond des ensembles de données, vous n'avez pas besoin de le modifier.

```
var classification_training = data_stack.sampleRegions({
  collection: polygons,
  properties: ['classcode'],
  scale: 10,
  tileScale: 16
});
```

Ensuite, nous construisons le classificateur de random forest et l'entraînons avec les données d'entraînement que nous venons d'extraire. Le classificateur que nous créons sera appelé RF_classifier, 500 définit le nombre d'arbres à utiliser dans le classificateur de forêt aléatoire, classification_training est les données d'entraînement, et classcode spécifie les classes qui seront utilisées dans la classification. 500 arbres sont spécifiés ici, bien que si des erreurs de limite de mémoire sont rencontrées lors de l'exécution du code GEE, ce nombre peut être réduit.

```
var RF_classifier = ee.Classifier.smileRandomForest(500)
    .train(classification_training, 'classcode');
```

Maintenant que nous avons entraîné le classificateur, nous pouvons classer la pile de données. land_cover_classification est le nom de la classification de sortie, data_stack est la pile de données d'entrée qui sera classée, et RF_classifier est le classificateur de forêt aléatoire que nous venons d'entraîner.

```
var land_cover_classification = data_stack.classify(RF_classifier);
```

Nous pouvons ensuite ajouter la classification au panneau du visualiseur pour l'examiner, en affichant la classification à l'aide de la palette de couleurs définie précédemment. Map.addLayer() est la commande permettant d'afficher un objet dans le panneau de visualisation. land_cover_classification est l'objet qui sera affiché, palette : palette spécifie le nom de la palette de couleurs que nous souhaitons appliquer, et les valeurs min et max définissent la plage de valeurs des données à afficher. Ici, huit classes de couverture du sol sont présentes, les valeurs (1 à 8) désignant les différentes classes de couverture du sol. Map.centerObject() indique au panneau de visualisation de zoomer sur une boîte de délimitation de l'objet spécifié, ici l'étendue de la table avec le niveau de zoom défini sur 10.

```
Map.addLayer(land_cover_classification,{palette: palette, min:1,max:8});
Map.centerObject(table, 10)
```

En plus de générer la classification de l'occupation du sol, il est également nécessaire d'effectuer une évaluation de la précision pour en apprécier la qualité. Les classificateurs de forêt aléatoire peuvent produire des statistiques d'évaluation de la précision « out-of-bag », bien que celles-ci gonflent généralement trop la précision déclarée de la classification. Une alternative qui fournit généralement des chiffres de précision plus réalistes pour une classification, est d'effectuer une évaluation de la précision en utilisant un ensemble de données de validation indépendant. Le code des deux approches est présenté ici. Tout d'abord, nous allons effectuer une évaluation de la précision en utilisant l'approche de la forêt aléatoire « out-of-bag ». Les commandes suivantes génèrent et renvoient, dans cet ordre, une matrice d'erreurs (souvent appelée matrice de confusion/correspondance), la précision globale de la classification, la précision du producteur et de l'utilisateur pour chaque classe d'occupation du sol et le coefficient de Kappa. La commande print() renvoie le résultat de la commande dans l'onglet 'Console'. Dans le script d'accompagnement, ces commandes sont commentées mais sont incluses afin que l'utilisateur puisse mettre en œuvre l'évaluation de la précision « out-of-bag » s'il le souhaite.

```
var RF_classifier_rf_error_matrix = RF_classifier.confusionMatrix();
  print('RF error matrix: ', RF_classifier_rf_error_matrix);
var RF_classifier_rf_accuracy = RF_classifier.confusionMatrix().accuracy();
  print('RF accuracy: ', RF_classifier_rf_accuracy);
var RF_classifier_rf_producers_accuracy = RF_classifier.confusionMatrix().producersAccuracy();
  print('RF producers accuracy: ',RF_classifier_rf_producers_accuracy);
var RF_classifier_rf_users_accuracy = RF_classifier.confusionMatrix().consumersAccuracy();
  print('RF users accuracy: ',RF_classifier_rf_users_accuracy);
var RF_classifier_kappa = RF_classifier.confusionMatrix().kappa();
```

```
print('RF kappa: ',RF_classifier_kappa);
```

Ensuite, nous allons effectuer l'évaluation de la précision avec l'ensemble de données indépendant qui a été précédemment importé en tant que table3. Tout d'abord, nous extrayons les classes de couverture terrestre pour les emplacements des points de validation indépendants, en créant un objet appelé validation_extraction.

```
var validation_extraction = data_stack.sampleRegions({
  collection: table3,
  properties: ['classcode'],
  scale: 10,
  tileScale: 16
});
```

Ensuite, nous générons une matrice de confusion, la précision globale, la précision du producteur, la précision de l'utilisateur et le coefficient Kappa et nous les renvoyons dans l'onglet "Console".

```
var confusionMatrix = ee.ConfusionMatrix(validation_extraction.classify(RF_classifier)
    .errorMatrix({
      actual: 'classcode',
      predicted: 'classification'
    }));
print('Confusion matrix:', confusionMatrix);
print('Overall Accuracy:', confusionMatrix.accuracy());
print('Producers Accuracy:', confusionMatrix.producersAccuracy());
print('Users Accuracy:', confusionMatrix.consumersAccuracy());
print('Kappa:', confusionMatrix.kappa());
```

5.5.6 Calculer la proportion de chaque classe d'occupation du sol en utilisant une fenêtre mobile

Ensuite, nous allons calculer comment la proportion de chaque classe d'occupation du sol varie dans la zone d'étude. Pour ce faire, un processus en deux étapes est nécessaire. Premièrement, nous prenons la classification de la couverture terrestre et à partir de celle-ci, nous générons des couches binaires de présence/absence pour chaque classe de couverture terrestre individuellement. Ensuite, nous utilisons une fenêtre mobile avec une taille de noyau correspondant à la distance de vol *d'An. gambiae* (846m) pour calculer la proportion occupée par la classe en question dans la fenêtre. La fenêtre mobile effectue ce calcul pour chaque pixel dans les couches binaires de présence/absence - pour un pixel donné, elle crée un tampon de 846m de

rayon autour de ce pixel, calcule la zone où la couverture terrestre en question est présente, puis convertit cette valeur en une proportion de la fenêtre. La fenêtre mobile passe ensuite au pixel suivant et répète le processus, et ainsi de suite jusqu'à ce que le calcul ait été effectué sur chaque pixel de la couche binaire de présence/absence pour une classe d'occupation du sol donnée.

La première étape consiste à "remapper" les valeurs du raster de la classification de l'occupation du sol (où différentes valeurs entières de pixel correspondent à différentes classes d'occupation du sol) en valeurs binaires où "1" correspond à la classe d'occupation du sol concernée, et où toutes les autres valeurs (les autres classes d'occupation du sol) reçoivent la valeur "0". Dans le code ci-dessous, nous effectuons cette opération pour la classe de couverture végétale de la forêt, qui a le code de classe de couverture végétale '1'. Ici, la couche binaire de présence/absence de sortie qui sera générée sera appelée `class1_only`. Dans la classification `land_cover_classification` d'entrée, huit classes de couverture du sol sont présentes, avec les codes de classe correspondants 1, 2, 3, 4, 5, 6, 7 et 8. Comme nous ne sommes intéressés que par la classe 1 (forêt), nous remappons les valeurs des classes à 1 (forêt de retenue), puis 0, 0, 0, 0, 0, 0, 0 pour mettre les valeurs de toutes les autres classes à 0.

```
var class1_only = land_cover_classification.remap([1,2,3,4,5,6,7,8],[1,0,0,0,0,0,0,0]);
```

Si nous voulions répéter le processus pour les prairies (code de classe de couverture terrestre 2), nous adapterions le code comme suit

```
var class2_only = land_cover_classification.remap([1,2,3,4,5,6,7,8],[0,1,0,0,0,0,0,0]);
```

Dans le script GEE ci-joint, cette opération est effectuée pour chaque classe d'occupation du sol, mais par souci de concision, nous ne présentons ici que ces exemples. Nous effectuons ensuite le calcul de la fenêtre mobile pour générer la proportion de la classe d'occupation du sol en question (ici la forêt, classe 1). Ceci produit l'objet de sortie `class1_mean_mw`, spécifie `class1_only` comme trame d'entrée, `focal_mean()` est la commande pour exécuter la fenêtre mobile, `flight_distance` correspond à la distance de vol de 846m d'*An. gambiae* (ceci a été spécifié plus tôt dans le script), `'circle'` définit un noyau circulaire (plutôt que carré) à utiliser, et `'meters'` spécifie que la distance de vol définie est en mètres, plutôt qu'en pixels. Encore une fois, dans le script GEE, ceci est effectué pour chaque classe de couverture terrestre, bien qu'un seul exemple soit présenté ici.

```
var class1_mean_mw = class1_only.focal_mean(flight_distance,'circle','meters');
```

5.5.7 Calculer les rasters 'distance à'

Ensuite, nous allons générer d'autres couches de données matricielles (rasters) donnant la distance entre un pixel et la parcelle la plus proche d'une classe

d'occupation du sol spécifique. Ceci est particulièrement intéressant pour les lisières de zones boisées (habitat de repos pour les moustiques) et les classes relatives à l'eau (habitat de reproduction potentiel). Le code ci-dessous génère un produit de distance pour la classe forêt, les valeurs des pixels de sortie étant la distance entre l'emplacement de ce pixel et la parcelle boisée la plus proche en mètres. Dans le script GEE qui l'accompagne, cette opération est également répétée pour les classes d'occupation du sol jachère, forêt et jachère combinées, eau courante et eau statique.

```
var dist_to_forest_m_int =
class1_only.fastDistanceTransform(500).sqrt().multiply(ee.Image.pixelArea().sqrt()).toInt();
```

5.5.8 Indices de végétation

Ensuite, nous générons une série de rasters d'indice de végétation (VI) et de teneur en eau (WI). Pour la zone d'étude. Les valeurs de l'indice de végétation et de l'indice de teneur en eau de la zone d'étude changeront tout au long de l'année en fonction de la phénologie de la végétation et des précipitations saisonnières. Ici, nous sommes intéressés par la génération de valeurs VI et WI "typiques" sur l'ensemble de l'année, plutôt que de calculer ces valeurs pour une date particulière dans l'année. Par conséquent, nous utilisons la collection d'images Sentinel-2 que nous avons créée précédemment et nous utilisons un réducteur 'médiane' pour calculer les valeurs médianes sur la période désignée. Deux VI, l'indice de végétation par différence normalisée (NDVI) et l'indice de végétation ajusté au sol (SAVI) seront produits, ainsi que deux WI, l'indice de teneur en eau par différence normalisée (NDWI) et l'indice de teneur en eau par différence normalisée modifié (MNDWI). Les fonctions permettant de générer ces produits VI et WI sont prédéfinies au début du script GEE, et ici nous mappons ces fonctions sur la collection d'images `S2_bandsubset`. Nous calculons ensuite les valeurs médianes pour chaque VI et WI séparément.

```
var S2_NDVI_medx10k_int = S2_bandsubset.map(NDVI).median().multiply(10000).toInt();
var S2_SAVI_medx10k_int = S2_bandsubset.map(SAVI).median().multiply(10000).toInt();
var S2_NDWI_medx10k_int = S2_bandsubset.map(NDWI).median().multiply(10000).toInt();
var S2_MNDWI_medx10k_int = S2_bandsubset.map(MNDWI).median().multiply(10000).toInt();
```

Une fois les fonctions mappées, nous effectuons une multiplication x10000, puis convertissons les sorties résultantes en valeurs entières à l'aide de `toInt()`. Les données sont converties au format entier afin de réduire la taille des données créées (les données au format entier nécessitent moins de capacité de stockage que d'autres formats tels que le flottant ou le double), cependant pour les valeurs d'indice de végétation telles que NDVI qui ont fréquemment une plage de valeurs entre -1 et 1, la conversion directe au format entier peut tronquer la plage de données et entraîner une perte de données. Pour éviter cela, les valeurs des données sont multipliées par 10000 avant d'être converties en entier, préservant ainsi la plage de valeurs des données. Les utilisateurs doivent noter, lors de l'inspection ultérieure des données, qu'elles ont été multipliées par 10000 par rapport aux valeurs de données originales.

5.5.9 Extraction et lissage des données CHIRPS

Enfin, nous extrairons les données pluviométriques de la zone d'étude pour la période et les quelques mois précédant le début de la collecte des données sur les moustiques. Les précipitations sont utilisées ici comme un indicateur de la disponibilité de l'habitat de reproduction des moustiques, les précipitations générant des bassins éphémères qui sont fréquemment utilisés comme habitat de reproduction. L'ensemble de données sur les précipitations qui sera utilisé est le CHIRPS Daily : Climate Hazards Group InfraRed Precipitation with Station Data (version 2.0 finale), identifié dans GEE comme "UCSB-CHG/CHIRPS/DAILY".

Les données CHIRPS à utiliser ici - avec un ensemble de données CHIRPS pour chaque mois civil - sont importées en tant que ressources au début du script et ne nécessitent pas d'analyse supplémentaire dans ce script. Toutefois, si les utilisateurs souhaitent répéter ce traitement pour d'autres sites ou dates, les étapes de production de ces ensembles de données sont décrites ci-dessous. Cet ensemble de données a une résolution spatiale de 0,05 degré d'arc et peut présenter un effet de bord de pixel perceptible. Par conséquent, nous appliquons une étape de lissage aux données CHIRPS pour minimiser les effets de bord des pixels de ce jeu de données. Une des limites de cette analyse est la taille maximale du noyau de 512 pixels qui peut être utilisée pour ce lissage, ce qui, à une résolution de 10 m, est une taille de noyau insuffisamment petite pour effectuer ce lissage. Pour contourner cette limitation, les données CHIRPS peuvent être traitées séparément et exportées vers des ressources avec une taille de pixel de 20m (au lieu de 10m), ce qui augmente suffisamment la taille du noyau.

Les commandes ci-dessous génèrent un jeu de données CHIRPS mensuel pour le mois calendaire d'août 2014 nommé `Lodja_mean_2014_aug_smoothx10k_int`. Une série de commandes est appliquée séquentiellement ici. `ee.ImageCollection('UCSB-CHG/CHIRPS/DAILY')` spécifie l'identifiant approprié du jeu de données GEE à partir duquel nous allons générer une collection. Les données CHIRPS sont un produit quotidien, mais pour des raisons de cohérence avec les périodes de collecte de données mensuelles sur les moustiques, nous les convertirons en un produit de précipitations moyennes mensuelles. Cette collection est ensuite filtrée par date pour retenir les données du mois calendaire d'août 2014 en utilisant la commande `.filter(ee.Filter.date('2014-08-01', '2014-08-31'))` avant de générer une valeur quotidienne moyenne pour cette période `.reduce(ee.Reducer.mean())`. La commande suivante `.focal_mean(5000, 'circle', 'meters')` ; effectue le lissage par fenêtre mobile, en utilisant une fenêtre mobile circulaire de 5000 mètres de rayon. Celle-ci est ensuite multipliée par 10000 et convertie au format entier.

```
var Lodja_mean_2014_aug_smoothx10k_int = ee.ImageCollection('UCSB-
CHG/CHIRPS/DAILY').filter(ee.Filter.date('2014-08-01', '2014-08-
31')).reduce(ee.Reducer.mean()).focal_mean(5000,'circle','meters').multiply(10000).toInt();
```

L'objet `Lodja_mean_2014_aug_smoothx10k_int` sera ensuite exporté vers des ressources en utilisant le code ci-dessous. Ici `image :` correspond à l'objet à exporter, `description :` est le nom du fichier de sortie, `scale :` est la résolution spatiale du fichier de sortie (c'est ici que nous spécifions 20m) `maxPixels :` définit le nombre maximum de pixels qui peuvent être

exportés, region : définit l'étendue pour laquelle l'analyse doit être exécutée et les données exportées (définie ici à la zone d'intérêt définie dans le polygone 'table' qui a été précédemment importé comme une ressource), et crs : spécifie le système de coordonnées / projection du fichier de sortie.

```
Export.image.toAsset({
 image: Lodja_mean_2014_aug_smoothx10k_int,
 description: Lodja_mean_2014_aug_smoothx10k_int,
 scale: 20,
 maxPixels: 1e13,
 region: table,
 crs: "EPSG:4326"
 });
```

Cette opération sera répétée pour chaque mois civil requis. Ces ensembles de données seront ensuite relus en tant que ressources, empilés avec les autres ensembles de données de variables environnementales générés ici, et exportés sous forme de pile à une résolution de 10 m afin que toutes les couches de la pile aient une résolution spatiale et une projection cohérentes.

5.5.10 Mettre à l'échelle et convertir les données en nombre entier, compiler les bandes et exporter

Nous avons maintenant généré toutes les couches de données dont nous avons besoin à partir de GEE. Ensuite, nous compilons les bandes dont nous avons besoin pour l'analyse ultérieure et exportons les données vers les ressources (Asset) sous forme de raster multi-bandes, chaque bande de la pile correspondant à une variable explicative. Il est avantageux d'exporter toutes les données de cette manière (par opposition à l'exportation de chaque couche individuellement), car cela garantit que chaque couche a la même étendue spatiale, la même taille de pixel, le même nombre de lignes et de colonnes de pixels et le même système de coordonnées.

Tout d'abord, nous convertissons toutes les variables qui ne sont pas déjà au format entier en nombres entiers, y compris une étape de multiplication par 10000 pour préserver les plages de données lorsque cela est nécessaire. Cette opération est effectuée pour les ensembles de données de la fenêtre mobile de chaque classe d'occupation du sol, avec un exemple donné ci-dessous pour la classe 1 (forêt). L'utilisateur doit être conscient que certaines couches de données seront donc 10000x les valeurs de données originales dans le jeu de données exporté.

```
var class1_mean_mwx10k_int = class1_mean_mw.multiply(10000).toInt();
```

Ensuite, nous compilons les bandes nécessaires à l'exportation dans un nouvel objet image appelé GEE_data_for_exportx10k_int. Nous affichons ensuite les noms des bandes dans GEE_data_for_exportx10k_int pour vérifier qu'elles sont toutes présentes.

```
var GEE_data_for_exportx10k_int =
class1_mean_mwx10k_int.addBands(class2_mean_mwx10k_int).addBands(class3_mean_mwx10k_int).addBands(class4_mean_mwx10k_int).addBands(class5_mean_mwx10k_int).addBands(class6_mean_mwx10k_int).addBands(class7_mean_mwx10k_int).addBands(class8_mean_mwx10k_int).addBands(elevation_int).addBands(aspect_int).addBands(slopex10k_int).addBands(tpi_15_pixel_int).addBands(dist_to_forest_m_int).addBands(dist_to_fallow_m_int).addBands(dist_all_forest_and_fallow_classes_m_int).addBands(dist_to_flowing_water_m_int).addBands(dist_to_static_water_m_int).addBands(S2_NDVI_medx10k_int).addBands(S2_SAVI_medx10k_int).addBands(S2_NDWI_medx10k_int).addBands(S2_MNDWI_medx10k_int).addBands(Lodja_mean_2014_aug_smoothx10k_int).addBands(Lodja_mean_2014_sept_smoothx10k_int).addBands(Lodja_mean_2014_oct_smoothx10k_int).addBands(Lodja_mean_2014_nov_smoothx10k_int).addBands(Lodja_mean_2014_dec_smoothx10k_int).addBands(Lodja_mean_2015_jan_smoothx10k_int).addBands(Lodja_mean_2015_feb_smoothx10k_int).addBands(Lodja_mean_2015_mar_smoothx10k_int).addBands(Lodja_mean_2015_apr_smoothx10k_int).addBands(Lodja_mean_2015_may_smoothx10k_int).addBands(Lodja_mean_2015_june_smoothx10k_int).addBands(Lodja_mean_2015_july_smoothx10k_int).addBands(Lodja_mean_2015_aug_smoothx10k_int).addBands(Lodja_mean_2015_sept_smoothx10k_int).addBands(Lodja_mean_2015_oct_smoothx10k_int).addBands(Lodja_mean_2015_nov_smoothx10k_int).addBands(Lodja_mean_2015_dec_smoothx10k_int).addBands(Lodja_mean_2016_jan_smoothx10k_int).addBands(Lodja_mean_2016_feb_smoothx10k_int).addBands(Lodja_mean_2016_mar_smoothx10k_int).addBands(Lodja_mean_2016_apr_smoothx10k_int).addBands(Lodja_mean_2016_may_smoothx10k_int).addBands(Lodja_mean_2016_june_smoothx10k_int).addBands(Lodja_mean_2016_july_smoothx10k_int).addBands(Lodja_mean_2016_aug_smoothx10k_int).addBands(Lodja_mean_2016_sept_smoothx10k_int).addBands(Lodja_mean_2016_oct_smoothx10k_int).addBands(Lodja_mean_2016_nov_smoothx10k_int).addBands(Lodja_mean_2016_dec_smoothx10k_int);

print(GEE_data_for_exportx10k_int.getInfo());
```

Ensuite, nous renommons les bandes individuelles avec des noms plus intuitifs.

```
var GEE_data_for_exportx10k_int = GEE_data_for_exportx10k_int.select(['remapped', 'remapped_1',
'remapped_2', 'remapped_3', 'remapped_4', 'remapped_5', 'remapped_6', 'remapped_7', 'elevation',
'aspect', 'slope', 'elevation_1', 'distance', 'distance_1', 'distance_2', 'distance_3', 'distance_4', 'B8',
'constant', 'B3', 'B3_1', 'precipitation_mean', 'precipitation_mean_1', 'precipitation_mean_2',
'precipitation_mean_3', 'precipitation_mean_4', 'precipitation_mean_5','precipitation_mean_6',
'precipitation_mean_7', 'precipitation_mean_8', 'precipitation_mean_9', 'precipitation_mean_10',
'precipitation_mean_11', 'precipitation_mean_12', 'precipitation_mean_13', 'precipitation_mean_14',
'precipitation_mean_15','precipitation_mean_16', 'precipitation_mean_17', 'precipitation_mean_18',
'precipitation_mean_19', 'precipitation_mean_20', 'precipitation_mean_21', 'precipitation_mean_22',
'precipitation_mean_23', 'precipitation_mean_24', 'precipitation_mean_25', 'precipitation_mean_26',
'precipitation_mean_27','precipitation_mean_28'],

['proportion_forest', 'proportion_grassland', 'proportion_clearing', 'proportion_fallow',
'proportion_built_up', 'proportion_flowing_water', 'proportion_static_water', 'proportion_burnt',
'elevation', 'aspect', 'slope', 'TPI', 'distance_to_forest', 'distance_to_fallow',
'distance_to_forest_or_fallow', 'distance_to_flowing_water', 'distance_to_static_water', 'Median_NDVI',
'Median_SAVI', 'Median_NDWI', 'Median_MNDWI','CHIRPS_Aug_2014', 'CHIRPS_Sept_2014',
'CHIRPS_Oct_2014', 'CHIRPS_Nov_2014', 'CHIRPS_Dec_2014', 'CHIRPS_Jan_2015', 'CHIRPS_Feb_2015',
'CHIRPS_Mar_2015', 'CHIRPS_Apr_2015', 'CHIRPS_May_2015', 'CHIRPS_June_2015',
'CHIRPS_July_2015','CHIRPS_Aug_2015', 'CHIRPS_Sept_2015', 'CHIRPS_Oct_2015', 'CHIRPS_Nov_2015',
'CHIRPS_Dec_2015', 'CHIRPS_Jan_2016','CHIRPS_Feb_2016', 'CHIRPS_Mar_2016', 'CHIRPS_Apr_2016',
```

```
'CHIRPS_May_2016', 'CHIRPS_June_2016', 'CHIRPS_July_2016', 'CHIRPS_Aug_2016',
'CHIRPS_Sept_2016', 'CHIRPS_Oct_2016', 'CHIRPS_Nov_2016', 'CHIRPS_Dec_2016']);
```

Nous affichons ensuite à nouveau les noms des bandes dans GEE_data_for_exportx10k_int pour vérifier qu'ils ont été modifiés.

```
print(GEE_data_for_exportx10k_int.getInfo());
```

Nous exportons ensuite la pile de données vers les ressources:

```
Export.image.toAsset({
 image: GEE_data_for_exportx10k_int,
 description: 'GEE_data_for_exportx10k_int',
 scale: 10,
 maxPixels: 1e13,
 region: table,
 crs: "EPSG:4326"
 });
```

5.5.11 Importer les données d'échantillonnage des moustiques et extraire les données des bandes

Maintenant que nous avons créé la pile de données comprenant les bandes de variables environnementales explicatives, nous devons extraire les valeurs de chaque bande pour le moment et le lieu où l'échantillonnage des moustiques a été effectué. Tout d'abord, nous devons effectuer une opération de filtrage pour prendre l'ensemble complet de données sur les moustiques et le sélectionner les séries d'ensembles de données spécifiques au mois. Nous procédons ainsi car nous ne souhaitons pas extraire les valeurs de précipitations pour chaque mois, mais seulement pour le mois civil au cours duquel chaque sous-ensemble de données sur les moustiques a été collecté, ainsi que pour les cinq mois précédents. Par conséquent, ces données doivent être extraites mois par mois. L'ensemble complet de données sur les moustiques a déjà été importé en tant que ressource nommée mosquito_survey_data. Le code suivant subdivise l'ensemble de données sur les moustiques, créant un nouveau sous-ensemble de données nommé mosquito_survey_data_2015_jan qui contient uniquement les données sur les moustiques collectées en janvier 2015.

```
var mosquito_survey_data_2015_jan =
mosquito_survey_data.filter(ee.Filter.eq("Year",2015)).filter(ee.Filter.eq("Month",'January'));
```

Cette opération effectue deux filtrages sur l'ensemble des données sur les moustiques, en filtrant d'abord par l'attribut " année " et en retenant tous les enregistrements collectés en 2015, puis en appliquant un second filtre pour ne retenir que les enregistrements identifiés comme " janvier " dans l'attribut " mois ". Ce processus est répété pour chaque mois civil entre janvier 2015 et décembre 2016.

Ensuite, nous créons une série de piles de données spécifiques au mois comprenant les bandes de variables non-CHIRPS et les bandes de précipitations CHIRPS pour le mois en question et les cinq mois précédents. Nous en extrayons les valeurs des bandes pour les sites d'échantillonnage des moustiques pour le mois en question. Tout d'abord, nous créons un nouvel objet nommé static_variables qui sélectionne toutes les bandes non-CHIRPS de la pile de données GEE_data_for_exportx10k_int.

```
var static_variables = GEE_data_for_exportx10k_int.select(['proportion_forest', 'proportion_grassland',
'proportion_clearing', 'proportion_fallow', 'proportion_built_up', 'proportion_flowing_water',
'proportion_static_water', 'proportion_burnt', 'elevation', 'aspect', 'slope', 'TPI', 'distance_to_forest',
'distance_to_fallow', 'distance_to_forest_or_fallow', 'distance_to_flowing_water',
'distance_to_static_water', 'Median_NDVI', 'Median_SAVI', 'Median_NDWI', 'Median_MNDWI']);
```

Ensuite, nous utilisons la commande ci-dessous pour créer de nouvelles piles de données comprenant la pile de données static_variables que nous venons de créer, y ajouter d'autres bandes de précipitations CHIRPS correspondant au mois en question et aux cinq mois précédents, puis appliquer une autre commande pour échantillonner les valeurs des bandes pour les lieux d'échantillonnage des moustiques pour ce mois donné. L'exemple ci-dessous concerne le mois de janvier 2015, et crée un nouvel objet contenant les données variables extraites, appelé training_2015_jan. Un seul exemple est présenté ici, mais dans le script GEE qui l'accompagne, cette opération est effectuée pour tous les mois de 2015 et 2016.

```
var training_2015_jan =
static_variables.addBands(Lodja_mean_2015_jan_smoothx10k_int).addBands(Lodja_mean_2014_dec_s
moothx10k_int).addBands(Lodja_mean_2014_nov_smoothx10k_int).addBands(Lodja_mean_2014_oct_
smoothx10k_int).addBands(Lodja_mean_2014_sept_smoothx10k_int).addBands(Lodja_mean_2014_aug
_smoothx10k_int).sampleRegions({collection: mosquito_survey_data_2015_jan,properties:
['An_gambiae'],  scale: 10});
```

Une fois cette opération effectuée pour chaque mois civil, les 24 ensembles de données variables extraites sont ensuite recombinés en un seul objet de collecte de caractéristiques.

```
var training_all_months =
training_2015_jan.merge(training_2015_feb).merge(training_2015_mar).merge(training_2015_apr).mer
ge(training_2015_may).merge(training_2015_june).merge(training_2015_july).merge(training_2015_au
g).merge(training_2015_sept).merge(training_2015_oct).merge(training_2015_nov).merge(training_201
5_dec).merge(training_2016_jan).merge(training_2016_feb).merge(training_2016_mar).merge(training
_2016_apr).merge(training_2016_may).merge(training_2016_june).merge(training_2016_july).merge(tr
aining_2016_aug).merge(training_2016_sept).merge(training_2016_oct).merge(training_2016_nov).mer
ge(training_2016_dec);
```

Nous exportons ensuite cet ensemble de données fusionnées afin que l'étape suivante de l'analyse, la sélection des caractéristiques à l'aide de la méthode Boruta, puisse être effectuée dans R. Les commandes suivantes exportent la collection de caractéristiques training_all_months dans un fichier au format csv nommé extracted_data et l'enregistrent dans le Google Drive de l'utilisateur.

```
Export.table.toDrive({
  collection: training_all_months,
  description:'extracted_data',
  fileFormat: 'CSV'
});
```

Depuis le Google Drive de l'utilisateur, ce fichier peut ensuite être téléchargé, enregistré localement et traité dans R.

5.6 Exécution du script

Cette section a jusqu'ici expliqué le contenu du script GEE, mais pas comment l'exécuter. Pour ce faire, il suffit de cliquer sur le bouton "Run" en haut à droite de la fenêtre GEE. Ceci exécutera le script complet, à l'exception du texte qui a été commenté. Certaines commandes écrivent les sorties dans l'onglet "Console", comme les résultats de l'évaluation de la précision, tandis que les tâches d'exportation vers les actifs ou les lecteurs apparaissent dans l'onglet "Tâches". Toutes les bandes et tous les ensembles de données ajoutés à la carte apparaîtront dans le visualiseur au bas de la fenêtre GEE.

Pour passer d'un onglet à l'autre, il suffit de cliquer sur celui qui vous intéresse. Vous pouvez également redimensionner les différents panneaux de la fenêtre GEE en cliquant simplement sur les barres qui les séparent et en les faisant glisser. Il est particulièrement utile d'agrandir la fenêtre de visualisation de la carte pour inspecter les couches que vous avez affichées. Vous devrez zoomer sur la carte pour voir la zone qui vous intéresse (par exemple, Lodja, RDC) (Figure 5.6), ce qui peut prendre un peu de temps pour le rendu.

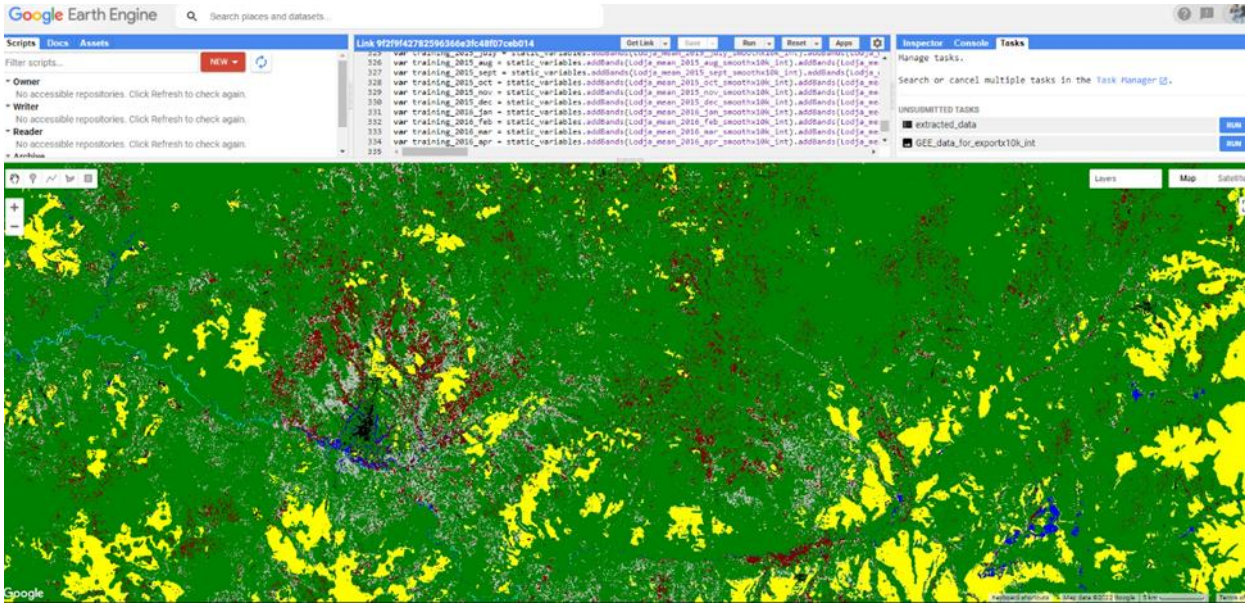


Figure 5.6 Visualisation de la carte de Lodja dans GEE avec la classification des terres.

Lorsque vous exécutez des tâches d'exportation vers un poste ou un lecteur, en plus d'exécuter la commande correspondante dans le script, vous devez également cliquer sur l'onglet "**Tasks**", où une liste des tâches d'exportation à exécuter est fournie. Il y aura un bouton '**Run**' à côté de chaque tâche - cliquez dessus, et la boîte '**Task : Initiate table export**' s'ouvre (Figure 5.7). Cliquez sur "Run" dans cette fenêtre et l'exportation commencera - ceci n'est pas fait automatiquement, donc si vous ne cliquez pas sur "**Run**", le processus d'exportation ne commencera pas. En fonction de la taille de l'objet à exporter et du volume des traitements effectués par d'autres utilisateurs sur GEE à ce moment-là, les exportations peuvent prendre un certain temps, voire des heures, pour se terminer. Il est toutefois possible d'exécuter plusieurs tâches simultanément et de poursuivre le traitement pendant que les tâches d'exportation précédentes sont encore en cours d'exécution en arrière-plan.

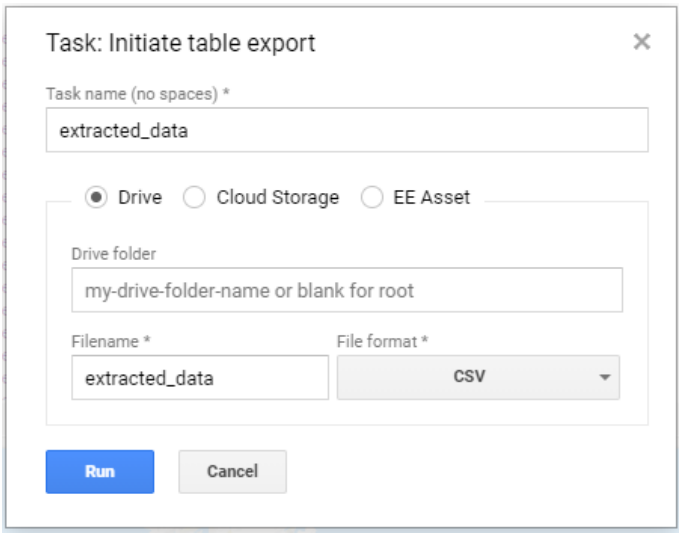


Figure 5.7 Fenêtre de lancement de l'exportation du tableau.

Notez que ce script implique un grand volume de traitement et d'analyse de données, donc une fois le script lancé, il peut prendre quelques minutes pour se terminer. Il se peut que vous receviez une notification " Page non réactive " - si c'est le cas, ne vous inquiétez pas, cela signifie simplement que le traitement est toujours en cours. Soyez patient et laissez le temps au traitement de s'exécuter, la boîte d'avertissement disparaîtra une fois le traitement terminé.

6 R - sélection des variables caractéristiques

L'étape suivante de l'analyse prend les données des variables environnementales extraites de la pile de données générée dans Google Earth Engine, et effectue une analyse de sélection des caractéristiques à l'aide de la **méthode Boruta** afin d'identifier lesquelles de cette série large de variables sont importantes par rapport à l'abondance d'An. gambiae. Comme pour la section GEE, cette étape de l'analyse est accompagnée d'un script qui permettra à l'utilisateur d'effectuer les étapes de l'analyse dans RStudio.

Avant de commencer l'analyse R, nous allons créer un **répertoire de travail** dans lequel nous allons sauvegarder les données précédemment extraites et exportées de Google Earth Engine. Pour ce faire, **ouvrez l'explorateur de fichiers**, naviguez jusqu'à un emplacement approprié sur votre ordinateur et créez un nouveau répertoire en cliquant avec le bouton droit de la souris, en sélectionnant "**Nouveau**", puis "**Dossier**". Vous pouvez ensuite donner un nom approprié au dossier que vous avez créé. Bien que l'emplacement et le nom du répertoire que vous créez puissent varier, pour cet exercice, nous utiliserons un répertoire avec le chemin d'accès suivant : "**D:\R_mosquito_modelling**". Copiez les données à analyser et le script R que nous vous avons fourni dans ce dossier. Si l'emplacement et le nom de votre répertoire de travail diffèrent, le script R doit être adapté pour spécifier l'emplacement et le nom du répertoire de travail alternatif.

Ensuite, ouvrez **RStudio**. Vous devriez voir une fenêtre similaire à celle illustrée à la Figure 6.1.

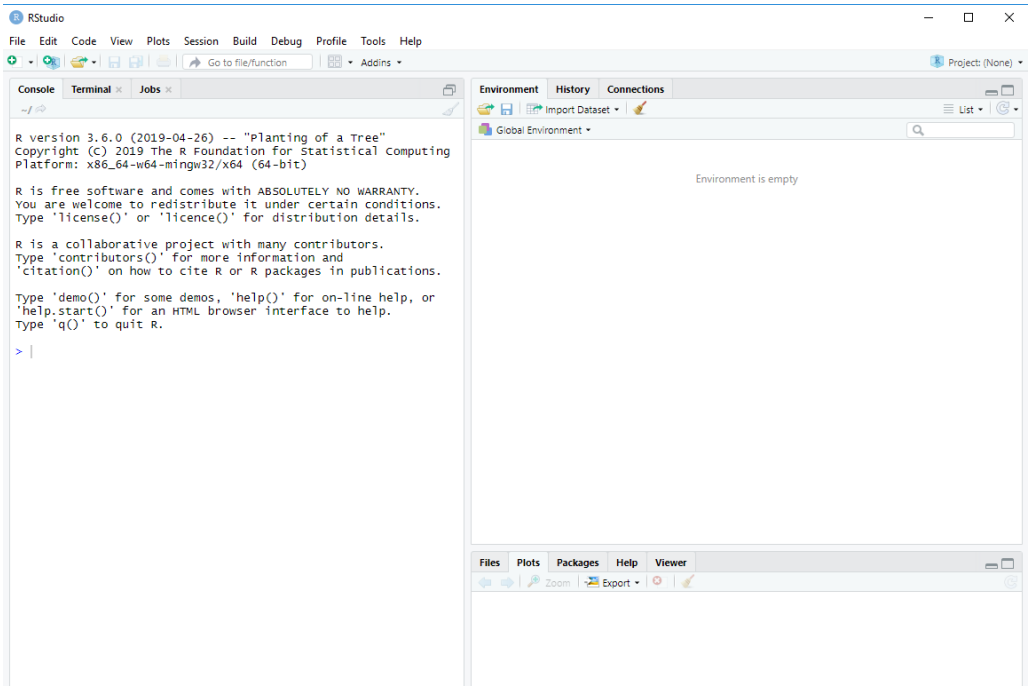


Figure 6.1. La fenêtre de RStudio

Si vous développez un nouveau script R, vous devez cliquer sur "**File**", puis "**New file**", puis "**R Script**", et une fenêtre supplémentaire s'ouvre en haut à gauche de RStudio, dans laquelle vous pouvez écrire votre code. Comme nous disposons d'un script déjà préparé pour effectuer cette analyse, nous pouvons simplement le charger. Encore une fois, allez dans '**File**', puis '**Open File**', et naviguez dans le répertoire de travail (D:\R_mosquito_modelling). Sélectionnez le fichier R contenant le script et cliquez sur "**Ouvrir**". Le script devrait maintenant s'afficher dans la fenêtre supérieure gauche de RStudio, comme dans la Figure 6.2.

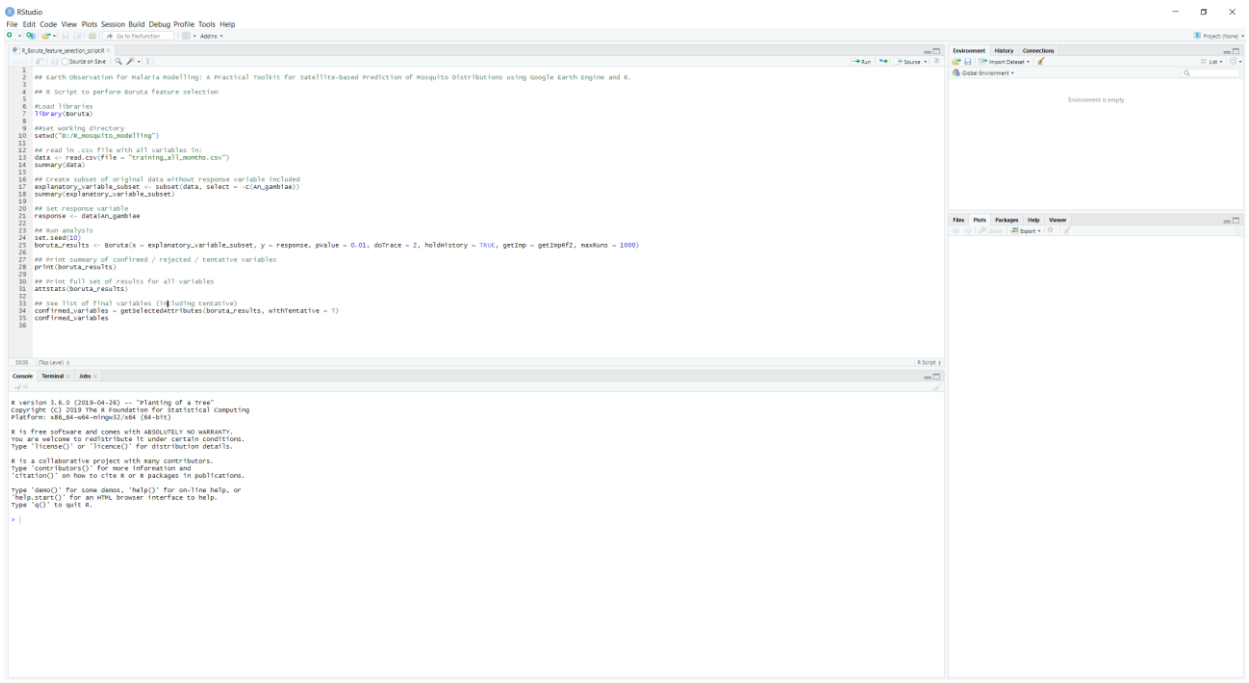


Figure 6.2. R Studio avec un exemple de script R chargé.

Comme dans GEE, des sections de commandes peuvent également être commentées dans un script RStudio, ces sections étant précédées de # et généralement affichées en vert.

Pour exécuter le script, ou une section du script, mettez d'abord en évidence la ou les lignes de code que vous souhaitez exécuter en cliquant sur la souris et en la faisant glisser. Une fois que la ou les sections correctes du texte ont été mises en évidence, cliquez sur l'icône Exécuter en haut à droite de la fenêtre dans laquelle le code est affiché. (ou Ctrl – Enter en raccourci clavier).

6.1 Charger les packages nécessaires et définir le répertoire de travail

R contient un large éventail de fonctionnalités, dont la plupart sont disponibles par le biais de paquets (packages) complémentaires. Ces paquets doivent être installés et appelés pour activer les fonctionnalités des commandes qu'ils contiennent. Les paquets ne doivent être installés qu'une seule fois mais devront être appelés au début

de chaque session RStudio. Les packages peuvent être installés manuellement via l'onglet "Packages" en bas à droite de la fenêtre RStudio (Figure 6.3).

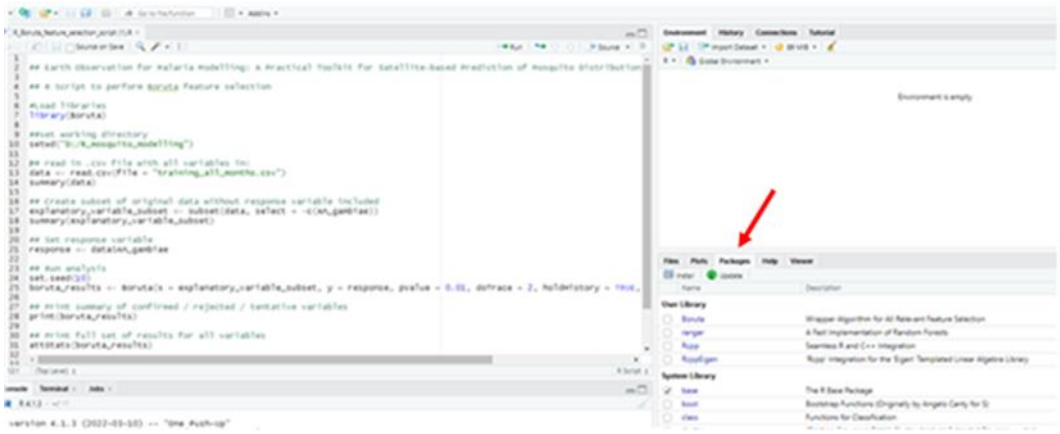


Figure 6.3 La flèche rouge indique où sélectionner les "paquets".

Cliquez sur l'onglet "Packages" si ce n'est pas déjà l'onglet actif. Ensuite, cliquez sur "Installer" et la fenêtre d'installation des paquets devrait s'ouvrir (Figure 6.4).

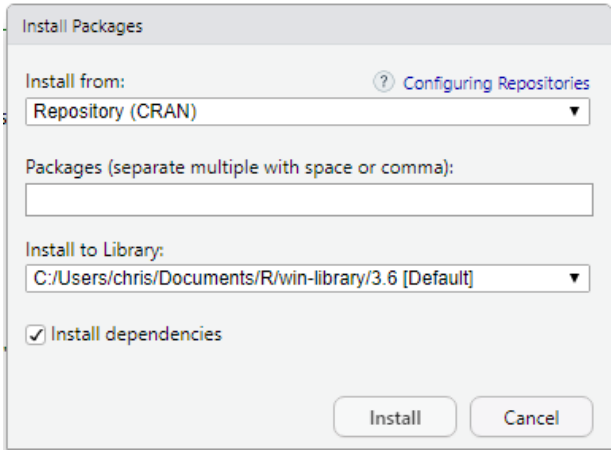


Figure 6.4. La fenêtre pop-up d'installation des paquets.

Dans cette fenêtre, tapez le nom du package que vous souhaitez installer dans la case "**Packages**". Au fur et à mesure que vous tapez, une liste déroulante des packages disponibles devrait apparaître. Sélectionnez le nom du package approprié, assurez-vous que la case "**Install dependencies**" est cochée, puis cliquez sur "**Install**". Répétez ce processus jusqu'à ce que tous les packages requis aient été installés. Procédez ainsi pour le paquet Boruta suivant.

Nous chargeons ensuite le paquet Boruta à l'aide de la commande library(). La commande library est répétée pour chacun des packages requis :

```
#Load libraries
library(Boruta)
```

Ensuite, nous indiquons le répertoire de travail créé précédemment, que R utilisera par défaut (sauf indication contraire) pour lire et écrire des données. Si l'emplacement et le nom du répertoire de travail diffèrent de ceux indiqués ici, vous devrez mettre à jour le code pour le refléter.

```
setwd("D:/R_mosquito_modelling")
```

Nous lisons ensuite le fichier .csv qui contient les données sur les moustiques et les variables environnementales que nous avons précédemment exportées de GEE. Nous utilisons la commande read.csv(), et lisons le fichier nommé training_all_months.csv qui est enregistré dans le répertoire de travail. Comme nous avons déjà spécifié le répertoire de travail, nous n'avons pas besoin d'inclure le chemin complet du fichier ici. La commande va lire le fichier csv et créer objet (de type data.frame) avec les données. Enfin, nous produisons un résumé des données que nous venons d'importer en utilisant la commande summary(). Celle-ci génère des statistiques sommaires de base pour chacune des variables de l'objet. Elles apparaissent dans la fenêtre de la console (Figure 6.5).

```
data <- read.csv(file = "training_all_months.csv")
summary(data)
```

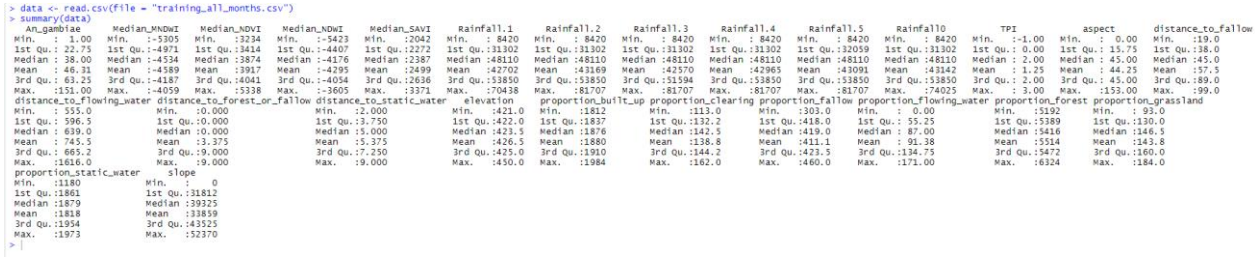


Figure 6.5. Statistiques récapitulatives des attributs des données.

Ensuite, nous devons créer un ensemble de données explanatory_variable_subset subset pour ne conserver que les variables explicatives. Pour cela, nous utilisons la commande subset, en spécifiant data comme l'ensemble de données source à partir duquel le sous-ensemble doit être créé, et nous utilisons la commande select = -c(An_gambiae) pour supprimer l'attribut An_gambiae. Nous utilisons ensuite à nouveau la fonction summary() pour le nouvel objet explanatory_variable_subset afin de vérifier que l'attribut An_gambiae a été supprimé.

```
explanatory_variable_subset <- subset(data, select = -c(An_gambiae))
```

```
summary(explanatory_variable_subset)
```

Nous créons ensuite un nouvel objet appelé réponse contenant uniquement la variable réponse An_gambiae.

```
response <- data$An_gambiae
```

Ensuite, nous exécutons l'analyse de sélection des caractéristiques via Boruta. Tout d'abord, nous définissons une valeur arbitraire de semence de 10. Nous spécifions que les résultats de sortie seront écrits dans un objet nommé boruta_results, nous spécifions que la commande Boruta() sera utilisée et nous spécifions les objets contenant les données des variables explicatives et des variables de réponse. Un certain nombre de paramètres internes sont ensuite spécifiés. La plupart d'entre eux n'ont pas besoin d'être modifiés, mais il est possible de changer la valeur de pValue utilisée, ainsi que le nombre d'itérations effectués dans l'analyse de sélection des caractéristiques. L'augmentation du nombre d'itérations peut être utile si les variables sont retournées comme "Tentatives", ce qui signifie que la sélection de caractéristiques ne peut pas confirmer si une variable est importante ou non. Ici, un nombre maximum de 1000 exécutions est spécifié.

```
set.seed(10)
boruta_results <- Boruta(x = explanatory_variable_subset, y = response, pValue = 0.01, doTrace = 2,
holdHistory = TRUE, getImp = getImpRfZ, maxRuns = 1000)
```

Au fur et à mesure que l'analyse se déroule, le texte est imprimé dans la fenêtre de la console. Le processus se termine soit lorsque toutes les variables ont été confirmées comme importantes ou rejetées, soit lorsque le nombre maximum d'exécutions spécifiées a été atteint. Lorsque le traitement est terminé, nous pouvons consulter les résultats pour voir si chaque variable explicative a été confirmée comme importante, non importante ou étiquetée comme hypothétique (considérées comme possibles, car proches du seuil de significativité). Nous pouvons afficher un résumé des résultats (Figure 6.6) à l'aide de la commande ci-dessous :

```
print(boruta_results)
```

```
> ## Print summary of confirmed / rejected / tentative variables
> print(boruta_results)
Boruta performed 999 iterations in 45.42988 secs.
 11 attributes confirmed important: distance_to_flowing_water, Median_NDWI, proportion_fallow, proportion_flowing_water, proportion_forest and 6 more;
 11 attributes confirmed unimportant: aspect, distance_to_fallow, distance_to_static_water, elevation, Median_MNDWI and 6 more;
 3 tentative attributes left: distance_to_forest_or_fallow, Median_NDVI, Median_SAVI;
>
```

Figure 6.6. Résultats de la sélection des caractéristiques via Boruta.

Le résumé des résultats sera imprimé dans la fenêtre de la console, et indiquera le nombre d'itérations effectuées et le temps d'exécution total de ces itérations, ainsi que le nombre d'attributs (variables) confirmés comme importants, non importants ou hypothétiques, et la liste de certaines de ces variables. Notez que si un grand nombre

de variables entre dans ces catégories, il se peut qu'elles ne soient pas toutes listées ici, et nous pouvons donc souhaiter consulter l'ensemble des résultats en utilisant la commande ci-dessous.

```
attStats(boruta_results)
```

Enfin, si nous souhaitons obtenir une liste des variables dont l'importance est confirmée, nous utilisons les commandes ci-dessous. Nous avons la possibilité d'inclure également les variables qui sont étiquetées comme hypothétiques en utilisant la commande withTentative = et en spécifiant soit T (inclure les variables hypothétiques) soit F (ne pas inclure les variables hypothétiques).

```
confirmed_variables = getSelectedAttributes(boruta_results, withTentative = T)
confirmed_variables
```

Les résultats de la sélection de caractéristiques via Boruta démontrent que pour l'abondance d'*An. gambiae*, dans ce cas, les variables du tableau 6.1 sont confirmées comme étant importantes, tandis que le reste des variables explicatives sont confirmées comme n'étant pas importantes et peuvent donc être écartées de la suite de l'analyse. Après avoir identifié cet ensemble parcimonieux de variables, nous retournons au GEE pour poursuivre les étapes de modélisation en nous concentrant uniquement sur ces variables.

Tableau 6.1. Les variables explicatives retenues après la sélection des caractéristiques de Boruta.

Variable explicative retenue
Couverture proportionnelle de la forêt
Couverture proportionnelle de la jachère
Couverture proportionnelle des eaux courantes
Couverture proportionnelle des eaux stagnantes
Distance jusqu'à la parcelle de forêt ou de jachère la plus proche
Distance jusqu'à la parcelle d'eau courante la plus proche
NDVI médian
SAVI médian
NDWI médian
Pluie0
Pluie-1
Pluie-2
Pluie-3
Pluie-4
Pluie-5

Ce processus de sélection des caractéristiques est présenté ici car les utilisateurs peuvent souhaiter adapter ce traitement à différentes espèces de moustiques dans différentes régions, où une série de variables environnementales différentes de celles identifiées ici peuvent exercer une influence plus grande sur l'abondance des moustiques.

7 Moteur Google Earth - modélisation

Le second script GEE exécute la modélisation Random Forest sur l'ensemble réduit de variables environnementales qui ont été confirmées comme importantes par rapport à l'abondance d'An. gambiae par l'analyse de sélection des caractéristiques via Boruta. Le script GEE pour cette étape de l'analyse est accessible via le lien ci-dessous :

https://code.earthengine.google.com/cd2b3c5d36dda2edb0efef5455ee58ba

Comme précédemment, un certain nombre de ressources sont déjà importées pour cette étape de l'analyse. De nouveau, nous importons la zone d'intérêt délimitant notre zone d'étude, les données de l'enquête sur les moustiques, les données de précipitations mensuelles lissées du CHIRPS d'août 2014 à décembre 2016, et la pile de données comprenant toutes les bandes de variables explicatives (GEE_data_for_exportx10k_int) que nous avons exportées comme ressource (asset) à la fin du premier script GEE.

Ensuite, un certain nombre de paramètres pour l'analyse de la forêt aléatoire sont définis : ntrees (le nombre d'arbres dans la forêt aléatoire), MinLeafPopulation (crée les nœuds dont l'ensemble d'entraînement contient au moins ce nombre de points), maxNodes (le nombre maximum de nœuds feuilles dans chaque arbre - si aucune limite n'est spécifiée, c'est la valeur par défaut), variablesPerSplit (le nombre de variables par division), et bagFraction (la fraction d'entrée dans le sac par arbre).

```
var ntrees = 200;
var MinLeafPopulation = 1;
var maxNodes = null;  // (no limit)
var variablesPerSplit = null;
var bagFraction = 0.7;
```

Nous prenons ensuite les données de l'enquête sur les moustiques qui ont été chargées en tant qu'actif, et nous sous-ensemblons à nouveau ces données dans une série d'ensembles de données spécifiques au mois, comme nous l'avons fait précédemment dans le premier script GEE. Ceci est fait pour chaque mois calendaire, avec le code d'exemple donné pour janvier 2015 ci-dessous.

```
var mosquito_survey_data_2015_jan =
mosquito_survey_data.filter(ee.Filter.eq("Year",2015)).filter(ee.Filter.eq("Month",'January'));
```

Ensuite, nous créons une pile de bandes matricielles statiques (sans précipitations) identifiées par l'analyse de sélection des caractéristiques comme étant importantes. Nous l'avons fait précédemment dans le premier script GEE, mais ici nous utiliserons

un ensemble réduit de variables statiques, en ignorant celles que l'analyse de sélection des caractéristiques a confirmé comme étant sans importance. Nous nommons la nouvelle pile de données static_variables et sélectionnons les bandes d'intérêt dans la pile de données GEE_data_for_exportx10k_int que nous avons créée dans le premier script GEE et importée précédemment en tant que ressource.

```
var static_variables =
GEE_data_for_exportx10k_int.select(['proportion_forest','proportion_fallow','proportion_flowing_water
','proportion_static_water','distance_to_forest_or_fallow','distance_to_flowing_water','Median_NDVI','
Median_SAVI','Median_NDWI']);
```

Nous créons ensuite une série de piles de données spécifiques au mois, composées de la pile de données static_variables que nous venons de créer et des bandes de données de précipitations CHIRPS pour le mois donné et les cinq mois précédents. Les bandes de précipitations ont déjà été importées en tant qu'actifs au début du script. L'exemple ci-dessous concerne le mois de janvier 2015, mais il est répété pour chaque mois civil entre janvier 2015 et décembre 2016.

```
var datastack_2015_jan =
static_variables.addBands(Lodja_mean_2015_jan_smoothx10k_int).addBands(Lodja_mean_2014_dec_s
moothx10k_int).addBands(Lodja_mean_2014_nov_smoothx10k_int).addBands(Lodja_mean_2014_oct_
smoothx10k_int).addBands(Lodja_mean_2014_sept_smoothx10k_int).addBands(Lodja_mean_2014_aug
_smoothx10k_int);
```

Alors que dans le premier script GEE, les piles de données similaires spécifiques au mois étaient utilisées pour extraire les valeurs variables des sites d'échantillonnage des moustiques en une seule étape, cette fois-ci, les piles de données servent à deux fins. Nous extrairons les emplacements des échantillons de moustiques comme auparavant, mais nous appliquerons également de manière prédictive les modèles de forêt aléatoire que nous générons sur ces piles de données pour produire des cartes d'abondance de moustiques prédites pour le mois en question sur toute l'étendue de la zone d'étude. Les commandes ci-dessus génèrent les piles de données spécifiques au mois dont nous avons besoin pour effectuer cette extrapolation prédictive. Ensuite, nous utilisons la commande sampleRegions() pour extraire les valeurs de toutes les bandes matricielles pour les emplacements des échantillons de moustiques. Cette opération est à nouveau exécutée pour chaque mois calendaire avec la commande pour janvier 2015 donnée ci-dessous.

```
var training_2015_jan = datastack_2015_jan.sampleRegions({collection:
mosquito_survey_data_2015_jan,properties: ['An_gambiae'],  scale: 10});
```

Ceci crée l'objet training_2015_jan, spécifie que le datastack_2015_jan est la pile de données pour extraire les valeurs des variables, que mosquito_survey_data_2015_jan est l'ensemble de données de l'enquête sur les moustiques pour ce mois pour lequel les emplacements d'échantillonnage seront utilisés pour extraire les données, properties : ['An_gambiae'] indique que l'attribut An_gambiae dans mosquito_survey_data_2015_jan (qui est le nombre d'*An.*

gambiae pour l'emplacement de l'enquête en question) doit également être inclus dans le jeu de données de sortie, et l'échelle : 10 spécifie la résolution spatiale à laquelle effectuer l'échantillonnage (ici, correspondant à la résolution de 10 m de la pile de données). Nous fusionnons ensuite les données d'entraînement extraites pour chaque mois en un seul objet appelé training_all_months en utilisant la commande ci-dessous.

```
var training_all_months =
training_2015_jan.merge(training_2015_feb).merge(training_2015_mar).merge(training_2015_apr).merge(training_2015_may).merge(training_2015_june).merge(training_2015_july).merge(training_2015_aug).merge(training_2015_sept).merge(training_2015_oct).merge(training_2015_nov).merge(training_2015_dec).merge(training_2016_jan).merge(training_2016_feb).merge(training_2016_mar).merge(training_2016_apr).merge(training_2016_may).merge(training_2016_june).merge(training_2016_july).merge(training_2016_aug).merge(training_2016_sept).merge(training_2016_oct).merge(training_2016_nov).merge(training_2016_dec);
```

Ensuite, nous identifions les noms de bande des piles de données mensuelles qui sont un paramètre requis de la modélisation de la forêt aléatoire. Nous extrayons les noms de bande en utilisant la commande ci-dessous pour la pile de données datastack_2015_jan. Les noms de bande doivent être identiques pour chaque pile de données mensuelles, nous n'avons donc besoin de le faire qu'une seule fois ici.

```
var bandNames = datastack_2015_jan.bandNames();
```

Nous construisons ensuite le modèle de forêt aléatoire. Les commandes créent un modèle nommé rf_regression, qui exécute la commande ee.Classifier. smileRandomForest qui lit les paramètres ntrees, variablesPerSplit, MinLeafPopulation, bagFraction et maxNodes définis plus tôt dans le script, spécifie le mode de sortie comme étant la régression (plutôt que la classification), définit training_all_months comme étant l'ensemble de données d'entraînement sur lequel construire le modèle de régression de la forêt aléatoire, An_gambiae comme étant la variable de réponse, et les bandNames comme étant les noms des variables explicatives que nous venons d'extraire de la pile de données (ci-dessus).

```
var rf_regression  =
ee.Classifier.smileRandomForest(ntrees,variablesPerSplit,MinLeafPopulation,bagFraction,maxNodes).setOutputMode('REGRESSION').train(training_all_months,"An_gambiae",bandNames);
```

Ce modèle est ensuite appliqué de manière prédictive sur chaque pile de données mensuelles, produisant un seul raster en sortie pour chaque mois. L'exemple ci-dessous concerne le mois de janvier 2015, mais il est répété pour chaque mois en modifiant la pile de données d'entrée et le nom du raster en conséquence. Ici, le nom de sortie pour le raster prédit est rf_2015_jan_predict, datastack_2015_jan est la pile de données d'entrée, et .classify() est la commande pour exécuter le modèle rf_regression de manière prédictive.

```
var rf_2015_jan_predict = datastack_2015_jan.classify(rf_regression);
```

Ensuite, nous empilons toutes les bandes mensuelles prédites dans une seule pile de données nommée RF_predicted et la sauvegardons dans un fichier. Ce sera ensuite l'entrée du script GEE final qui sera utilisé pour visualiser les cartes d'abondance prédite d'*An. gambiae*.

```
var RF_prediced =
rf_2015_jan_predict.addBands(rf_2015_feb_predict).addBands(rf_2015_mar_predict).addBands(rf_201
5_apr_predict).addBands(rf_2015_may_predict).addBands(rf_2015_june_predict).addBands(rf_2015_jul
y_predict).addBands(rf_2015_aug_predict).addBands(rf_2015_sept_predict).addBands(rf_2015_oct_pre
dict).addBands(rf_2015_nov_predict).addBands(rf_2015_dec_predict).addBands(rf_2016_jan_predict).a
ddBands(rf_2016_feb_predict).addBands(rf_2016_mar_predict).addBands(rf_2016_apr_predict).addBan
ds(rf_2016_may_predict).addBands(rf_2016_june_predict).addBands(rf_2016_july_predict).addBands(rf
_2016_aug_predict).addBands(rf_2016_sept_predict).addBands(rf_2016_oct_predict).addBands(rf_201
6_nov_predict).addBands(rf_2016_dec_predict);


Export.image.toAsset({
  image: RF_prediced,
  description: 'RF_prediced',
  scale: 10,
  maxPixels: 1e13,
  region: table,
  crs: "EPSG:4326"
  });
```

Enfin, nous exportons RF_predicted vers le drive (le Google Drive de l'utilisateur enregistré), à partir duquel la pile de données peut être téléchargée et chargée dans des logiciels SIG tels que QGIS.

```
Export.image.toDrive({
  image: RF_prediced,
  description: 'RF_prediced',
  scale: 10,
  maxPixels: 1e13,
  region: table,
  crs: "EPSG:4326"
  });
```

8 Google Earth Engine - visualisation des données

Pour visualiser les données prédites d'*An. gambiae*, le troisième script GEE a été développé pour afficher les ensembles de données mensuelles et permettre le partage des données dans une application GEE. Le lien pour ce script est ci-dessous :

https://code.earthengine.google.com/6b0fac43dcd7096b2cd342e44227614d

Ce script lit dans les ressources de la pile de données RF_predicted comprenant les abondances prédites d'An. gambiae pour chaque mois calendaire. La première section du code renomme les bandes pour leur donner des noms plus intuitifs, en précisant le mois et l'année auxquels elles correspondent.

```
var RF_predicted =
RF_predicted.select(['classification','classification_1','classification_2','classification_3','classification_4','classification_5','classification_6','classification_7','classification_8',‘classification_9','classification_10','classification_11','classification_12','classification_13','classification_14','classification_15','classification_16','classification_17','classification_18','classification_19','classification_20','classification_21','classification_22','classification_23'],

['January_2015_pred','February_2015_pred','March_2015_pred','April_2015_pred','May_2015_pred','June_2015_pred','July_2015_pred','August_2015_pred','September_2015_pred','October_2015_pred','November_2015_pred','December_2015_pred','January_2016_pred','February_2016_pred','March_2016_pred','April_2016_pred','May_2016_pred','June_2016_pred','July_2016_pred','August_2016_pred','September_2016_pred','October_2016_pred','November_2016_pred','December_2016_pred']);
```

Nous affichons ensuite l'abondance prédite pour chaque mois, avec un exemple de code pour janvier 2015 ci-dessous.

```
Map.addLayer(RF_predicted, {bands: ['January_2015_pred'], min:0 , max:100, palette: ['blue', 'green', 'red']}, "An gambiae predicted abundance January 2015", true);
```

Ici, la commande `Map.addLayer` affiche la bande spécifiée dans le visualiseur, `RF_predicted` spécifie la pile de données qui contient les données à afficher, et `['January_2015_pred']` spécifie la bande individuelle dans la pile de données `RF_predicted` que nous souhaitons afficher. `min:0 , max:100` spécifie la plage de données à afficher, et `palette : ['blue', 'green', 'red']` donne la palette de couleurs à utiliser pour afficher les données. Enfin, `"An gambiae predicted abundance January 2015"` donne l'étiquette à afficher pour cette couche, et true spécifie que la couche sera affichée automatiquement. Cette valeur peut également être définie sur false, auquel cas la bande ne sera pas affichée automatiquement, bien qu'elle puisse être affichée ultérieurement en cochant la case de cette couche qui apparaîtra si vous passez le curseur sur l'onglet Layers dans la fenêtre du visualiseur.

Il est utile, lorsque vous affichez un grand nombre de bandes, de définir initialement cette valeur sur false, car le rendu d'un grand nombre de bandes peut parfois prendre un certain temps. Toute bande que vous souhaitez afficher peut être activée à l'aide de sa case à cocher.

Enfin, nous utilisons la commande ci-dessous pour zoomer et centrer l'étendue du visualiseur sur `RF_predicted`.

```
Map.centerObject(RF_predicted);
```

Le reste du texte de commande dans le script correspond à la création d'une légende à afficher à côté des données que nous avons générées. Vous n'aurez pas besoin de modifier ce code et nous ne l'explorerons donc pas en détail ici.

L'exécution du script permet d'afficher les résultats dans le visualiseur de carte, mais l'idéal serait de pouvoir diffuser les résultats à d'autres personnes de manière pratique. GEE nous permet de le faire via sa fonctionnalité Apps, qui permet à d'autres personnes de visualiser les résultats de l'analyse via un navigateur web.

Pour publier le script en tant qu'application, cliquez sur le bouton "**Apps**" en haut à droite de l'écran de GEE 

. Dans la fenêtre "**Manage Apps**" qui s'ouvre, cliquez sur "**New App**" (en haut à droite), et la fenêtre "**Publish New App**" s'ouvre (Figure 8.1).

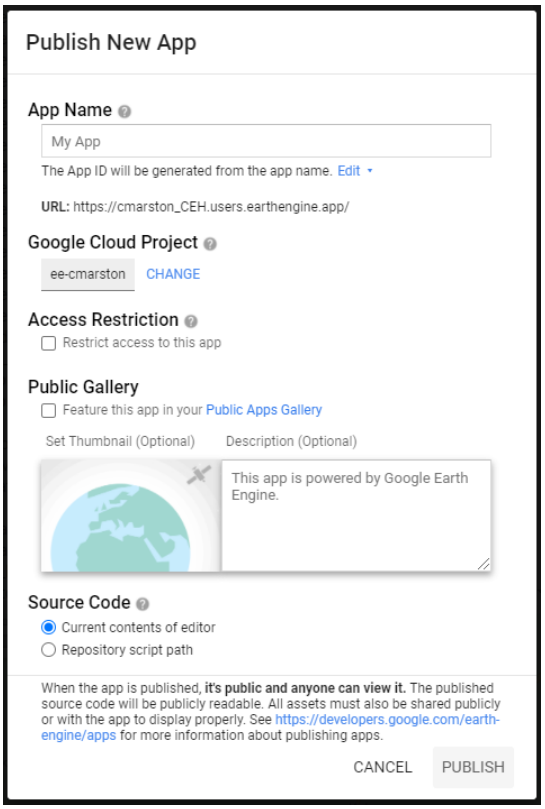


Figure 8.1. La fenêtre Publish New App.

Donnez à l'application un nom approprié, vérifiez que le contenu actuel de l'éditeur est coché sous "**Source Code**", puis cliquez sur "**Publish**". L'application apparaîtra alors dans la fenêtre "**Manage Apps**". Si vous apportez des modifications au script par la suite, vous devrez également mettre à jour l'application. Pour ce faire, dans la fenêtre "Gérer les applications", sous la colonne "**ID (clic to updtate app)**", cliquez sur le lien de l'application et la fenêtre "**App Details**" s'ouvre. Elle ressemble beaucoup à la fenêtre "**Publish New App**", mais sous "**Source Code**" en bas de cette fenêtre, cochez "**Current contents of editor**" puis "**Save**" et cela mettra à jour l'application pour inclure les changements apportés au code source.

Pour lancer l'application, cliquez sur le nom de l'application dans la colonne ' **App Name (click to launch)**', et l'application devrait ouvrir un navigateur Web et afficher les données comme le montre la Figure 8.2.

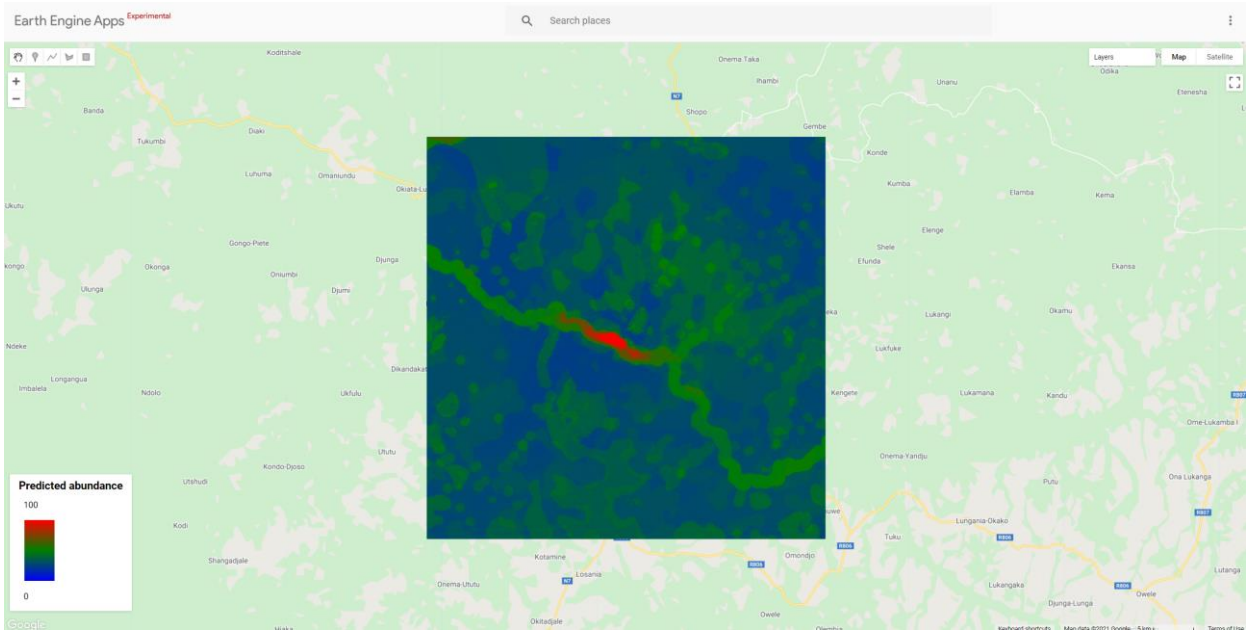


Figure 8.2. Application Google Earth Engine affichant les données sur l'abondance prédite des moustiques.

L'application affichera les données, mais ne donnera pas au spectateur l'accès aux ressources ou au code sous-jacent. En survolant le bouton "**Layers**" (en haut à droite), une liste déroulante apparaîtra avec toutes les couches disponibles pour l'affichage. La plupart d'entre elles sont actuellement configurées pour ne pas s'afficher, mais l'utilisateur peut utiliser les cases à cocher de chaque bande pour activer ou désactiver l'affichage de chacune d'entre elles à sa guise. Il peut également effectuer un zoom avant ou arrière et un panoramique autour de la zone d'étude, et aussi changer la couche de base de la carte affichée dans la Figure 8.2 pour une couche d'image satellite. Notez que la couche de base satellite n'est pas l'imagerie satellite utilisée dans le flux d'analyses que nous avons menées, mais un ensemble de données de plus haute résolution mis à disposition dans GEE pour le contexte / la visualisation mais pas l'analyse. Notez également qu'il y aura des décalages temporels entre les

dates d'acquisition de l'imagerie satellitaire que nous avons analysée, et les dates d'acquisition de la couche de base satellitaire.

Le lien de l'application, tel qu'il s'affiche dans le navigateur web, peut ensuite être distribué et visualisé par les utilisateurs qui n'ont alors besoin que d'une connexion web.

Notez que pour afficher les données de cette manière, les ressources qui sont chargées et affichées (ici la pile RF_predicted des bandes correspondant aux prédictions de la forêt aléatoire pour chaque mois calendaire) doivent être partagés. Pour ce faire, retournez à l'éditeur de code GEE contenant le script, cliquez sur l'onglet "**Assets**", puis cliquez à nouveau sur la ressource que vous souhaitez partager. Une nouvelle fenêtre semblable à celle de la Figure 8.3 devrait s'ouvrir.

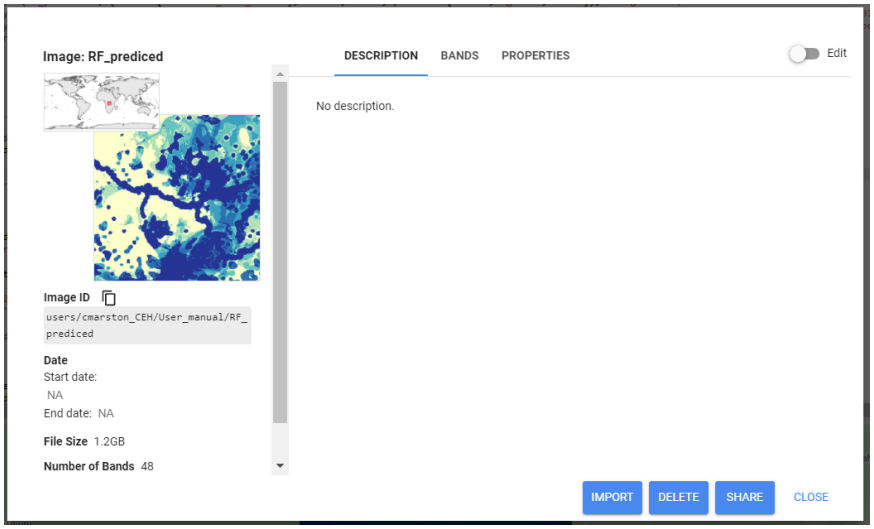


Figure 8.3. Fenêtre d'information sur les biens.

Dans cette fenêtre, cliquez sur "**Share**", cochez la case "**Anyone can read**", puis cliquez sur "**Done**". La ressource devrait maintenant être accessible à tous via l'application.

9 Décharge de responsabilité

Ce guide de l'utilisateur et les ensembles de données d'exemple sont fournis "tels quels" sans garantie d'aucune sorte, y compris, mais sans s'y limiter, les garanties implicites de qualité marchande et d'adéquation à un usage particulier. L'utilisateur assume l'entière responsabilité de l'exactitude et de l'adéquation de ce programme pour une application spécifique. En aucun cas, les auteurs ou les institutions affiliées ne pourront être tenus responsables de tout dommage, y compris les pertes de profits, les pertes d'économies ou tout autre dommage accessoire ou consécutif résultant de l'utilisation ou de l'impossibilité d'utiliser ce programme.

Les utilisateurs sont encouragés à utiliser ce matériel de formation et à l'adapter à leurs propres besoins. Si des publications sont développées en incorporant les méthodes présentées ici, ce guide de l'utilisateur et l'article de journal suivant à partir duquel il a été développé doivent être cités.

Ce guide de l'utilisateur doit être cité comme : Marston C.G., Rowland C.S., O'Neil A.W., Irish, S., Wat'senga F., Martin-Gallego P., Giraudoux P., et Strode C. 2022. Observation de la Terre pour la modélisation du paludisme : une boîte à outils pratique pour la prédiction par satellite de la distribution des moustiques à l'aide de Google Earth Engine et de R. UK Centre for Ecology and Hydrology.DOI : 10.5281/zenodo.7621047

L'article de journal devrait être cité comme : Marston C.G., Rowland C.S., O'Neil A.W., Irish, S, Wat'senga F., Martín-Gallego P., Aplin P., Giraudoux, P. and Strode, C. (2023). Developing the Role of Earth Observation in Spatio-Temporal Mosquito Modelling to Identify Malaria Hot-Spots. Remote Sensing. 15, 43, DOI:10.3390/rs15010043.

10 Remerciements

Ce projet a été financé par un fonds collectif UKRI administré par l'Université Edge Hill dans le cadre du Global Challenges Research Fund, par le Centre britannique d'écologie et d'hydrologie (numéro de projet NEC07217) et par le Natural Environment Research Council (numéro de bourse NE/R016429/1) dans le cadre du programme UK-SCAPE qui fournit une capacité nationale. Seth Irish a été soutenu par l'Initiative du Président des États-Unis contre le paludisme. Les résultats et les conclusions de ce document sont ceux des auteurs et ne représentent pas nécessairement la position officielle des Centers for Disease Control (CDC).

11 Références

Marston C.G., Rowland C.S., O'Neil A.W., Irish, S, Wat'senga F., Martín-Gallego P., Aplin P., Giraudoux, P. and Strode, C. (2023). Developing the Role of Earth Observation in Spatio-Temporal Mosquito Modelling to Identify Malaria Hot-Spots. Remote Sensing. 15, 43, DOI:10.3390/rs15010043.

Verdonschot, P. F., & Besse-Lototskaya, A. A. (2014). Flight distance of mosquitoes (Culicidae): a metadata analysis to support the management of barrier zones around rewetted and newly constructed wetlands. Limnologica, 45, 69-79. doi.org/10.1016/j.limno.2013.11.002

12 Glossaire

Couverture au sol

Désigne la couverture de surface du sol, qu'il s'agisse de végétation, d'infrastructures urbaines, d'eau, de sol nu ou autre.

Indice de teneur en eau par différence normalisée modifié (MNDWI)

Utilise les bandes vertes et infrarouges à ondes courtes (SWIR) pour quantifier les caractéristiques des eaux libres. Il minimise la détection des zones bâties qui sont souvent corrélées aux eaux libres dans d'autres indices.

Indice de végétation par différence normalisée (NDVI)

Quantifie la végétation en mesurant la différence entre le proche infrarouge (NIR) (que la végétation reflète fortement) et la lumière rouge (que la végétation absorbe).

Indice de teneur en eau par différence normalisée (NDWI)

Désigne l'un d'au moins deux indices dérivés de la télédétection liés à l'eau liquide ; soit les changements de la teneur en eau des feuilles en utilisant le NIR et le SWIR, soit les changements liés à la teneur en eau des masses d'eau, en utilisant les longueurs d'onde du vert et du NIR.

Polarisations

La polarisation est un moyen de donner aux signaux de transmission SAR une direction spécifique. Sentinel-1 peut émettre un signal en polarisation horizontale (H) ou verticale (V), puis le recevoir dans les deux polarisations H et V.

Rasters

Un raster consiste en une matrice de cellules (ou pixels) organisée en lignes et en colonnes (ou une grille) où chaque cellule contient une valeur représentant une information, telle que la réflectance de la surface. Les rasters sont constitués, entre autres, des photographies aériennes numériques, des images satellites, des images numériques ou même des cartes scannées.

Sentinel-1

Les satellites Sentinel-1A et Sentinel-1B partagent le même plan orbital. Tous deux utilisent un radar à synthèse d'ouverture (SAR) qui a l'avantage de fonctionner à des longueurs d'onde qui ne sont pas gênées par une couverture nuageuse ou un manque d'éclairage et qui peut acquérir des données au-dessus d'un site de jour comme de nuit dans toutes les conditions météorologiques. Sentinel-1, avec son capteur SAR en bande C, peut offrir une surveillance fiable et répétée d'une zone étendue.

Sentinel-2

Deux satellites SENTINEL-2 identiques fonctionnent simultanément, en phase à 180° l'un par rapport à l'autre, sur une orbite héliosynchrone à une altitude moyenne de 786 km. Sentinel-2 transporte le capteur MSI (Multispectral Instrument) qui acquiert des données dans 13 bandes spectrales avec quatre bandes à 10 m de résolution spatiale, six bandes à 20 m de résolution et trois bandes à 60 m de résolution.

Shapefile

Un format simple et non topologique pour le stockage de l'emplacement géographique d'objets et des informations sur les attributs des objets géoréférencés. Les caractéristiques géographiques dans un fichier de forme peuvent être représentées par des points, des lignes ou des polygones.

Indice de végétation ajusté au sol (SAVI)

Un indice de végétation qui tente de minimiser les influences de la luminosité du sol en utilisant un facteur de correction de la luminosité du sol. Il est souvent utilisé dans les régions arides où la couverture végétale est faible.

BANGOR
UK Centre for Ecology & Hydrology
Environment Centre Wales
Deiniol Road
Bangor
Gwynedd
LL57 2UW
United Kingdom
T: +44 (0)1248 374500
F: +44 (0)1248 362133

EDINBURGH
UK Centre for Ecology & Hydrology
Bush Estate
Penicuik
Midlothian
EH26 0QB
United Kingdom
T: +44 (0)131 4454343
F: +44 (0)131 4453943

LANCASTER
UK Centre for Ecology & Hydrology
Lancaster Environment Centre
Library Avenue
Bailrigg
Lancaster
LA1 4AP
United Kingdom
T: +44 (0)1524 595800
F: +44 (0)1524 61536

WALLINGFORD (Headquarters)
UK Centre for Ecology & Hydrology
Maclean Building
Benson Lane
Crowmarsh Gifford
Wallingford
Oxfordshire
0X10 8BB
United Kingdom
T: +44 (0)1491 838800
F: +44 (0)1491 692424

enquiries@ceh.ac.uk

www.ceh.ac.uk